

JAVASCRIPT IN URDU

جاوا اسکریپٹ
اردو زبان میں

☆☆☆ محمد دانش ارشاد ☆☆☆

JavaScript سب سے پہلے Netscape نے 1995 میں اپنے براؤزر کے لئے بنائی تھی۔ جس کو بعد میں تمام انٹر نیٹ براؤزرز نے استعمال کرنا شروع کر دیا۔ ابتداء میں JavaScript کا نام LiveScript تھا جو بعد میں تبدیل کر کے JavaScript رکھ دیا گیا۔ JavaScript ایک ہلکی اور آسان زبان ہے اور یہ ایک کمانڈ سائنڈ زبان ہے یعنی جس کے لئے کسی بھی قسم کی انسٹالیشن وغیرہ کی ضرورت نہیں ہوتی۔ PHP یا ASP وغیرہ کی طرح اس میں آپ کو کسی سرور کی ضرورت نہیں۔

آپ JavaScript کو کسی بھی ٹیکسٹ ایڈیٹر، مثلاً ++ Notepad, Notepad وغیرہ میں ٹائپ کر سکتے ہیں اور اپنے کسی بھی ویب براؤزر میں استعمال کر سکتے ہیں۔ JavaScript ایک Case-Sensitive زبان ہے مطلب ہے کہ اس کے کسی بھی کوڈ میں جو حرف چھوٹا ہوگا تو وہ آپ کو چھوٹا ہی لکھنا ہے اور جو حرف بڑا ہوگا وہ بڑا ہی لکھنا ہے۔ JavaScript کے اندر چھوٹے اور بڑے حرف دو الگ الگ چیزیں تصور کی جاتی ہیں۔

مثال کے طور پر Javascript اور javascript ان دونوں میں سے ایک کا "J" چھوٹا اور دوسرے کا بڑا ہے تو یہ دونوں الگ الگ الفاظ ہوں گے۔ آپ کو اس چیز کا خیال رکھنا ہوگا۔

JavaScript میں ہم کیا کیا کر سکتے ہیں؟

JavaScript کے اندر ہم مختلف کام کر سکتے ہیں۔

مثلاً:

ویب کی سیکورٹی، فارم ویلڈیشن، کوکی بنانا، یوزر کی ip، اسکا براؤزر، اسکا آپریٹنگ سسٹم کونسا ہے وغیرہ یوزر کو بتانا۔ ہم جاوا اسکرپٹ کے اندر گھڑی بنا سکتے ہیں۔ ویب پیج کے لئے چھوٹے موٹے گیمز وغیرہ بھی ڈیزائن کر سکتے ہیں۔

JavaScript سیکھنے کے لئے کیا ضروری ہے؟

JavaScript کے لئے آپ کے پاس HTML اور CSS کی بنیادی معلومات ضروری ہے۔ اگر آپ ان کے بارے



میں نہیں جانتے تو برائے مہربانی پہلے آپ ان دونوں کو سیکھیں۔

☆ JavaScript کے لئے آپ کو کمپیوٹر اور انٹرنیٹ کی بنیادی معلومات ہونی ضروری ہے۔

☆ ایک عدد ٹیکسٹ ایڈیٹر جیسے: Notepad, Notepad ++ وغیرہ ہونا لازمی ہے۔

☆ ایک عدد ویب براؤزر جیسے: Google Chrome, Mozilla Firefox, Internet Explorer, Safari, Opera, وغیرہ۔

JavaScript in Urdu کی اس کتاب میں ہم کیا کیا سیکھیں گے؟

ہم JavaScript in Urdu کی اس کتاب میں JavaScript کا بنیادی استعمال، اس کے کام کا طریقہ کار وغیرہ کے متعلق سیکھیں گے۔

فہرست

ترتیب	باب	صفحہ نمبر
1	JavaScript کا Syntax اور اس کا بنیادی استعمال	6
2	JavaScript کے Methods کون کون سے ہیں؟	8
3	JavaScript میں Comments داخل کرنا	8
4	JavaScript کے Data Types کیا ہیں؟	9
5	JavaScript میں Variables کی تخلیق اور ان کا استعمال کرنا	11
6	JavaScript میں Functions کی تخلیق اور ان کا استعمال کرنا	15
7	JavaScript میں String Method کا استعمال کرنا	18
8	JavaScript میں Number Methods کا استعمال کرنا	24
9	JavaScript میں Math Object کا استعمال کرنا	25
10	JavaScript میں Events کیا کیا ہیں اور ان کا استعمال کرنا	28
11	JavaScript کے Assignment Operators کیا ہیں؟	31
12	JavaScript کے Arithmetic Operator کیا ہیں؟	31
13	JavaScript کے Comparison Operators کیا ہیں؟	37
14	JavaScript کے Logical Operators کیا ہیں؟	44
15	JavaScript میں if Statement کا استعمال کرنا	44
16	JavaScript میں else Statement کا استعمال کرنا	45
17	JavaScript میں else if Statement کا استعمال کرنا	47
18	JavaScript میں Popup Boxes کیا کیا ہیں اور ان کا استعمال کرنا	48
19	JavaScript میں For Loop کا استعمال کرنا	50
20	JavaScript میں while کا استعمال کرنا	52
21	JavaScript میں do while کا استعمال کرنا	53
22	JavaScript میں Switch & Case Statement کا استعمال کرنا	55

59	JavaScript میں Return Statement کا استعمال کرنا	23
60	JavaScript میں Break & Continue Statement کا استعمال کرنا	24
63	JavaScript میں Type Conversion کا استعمال	25
74	JavaScript میں Date and Time کا استعمال کرنا	26
89	JavaScript میں Cookie تخلیق کرنا اور ان کا استعمال کرنا	27
91	JavaScript میں DOM کیا ہے؟	28
94	JavaScript میں BOM کیا ہے؟	29
109	JavaScript میں Console.log() کا استعمال کرنا	30
110	JavaScript میں Timing Events کا استعمال کرنا	31
112	JavaScript میں ایک گھڑی بنائیں گے۔	32
119	JavaScript میں ایک Calculator بنائیں گے۔	33
123	JavaScript میں ایک Temperature Convertor بنائیں گے۔	34

Syntax

سب سے پہلے ہم سیکھیں گے کہ JavaScript کو لکھتے کس طرح ہیں۔

JavaScript کو لکھنے کے لئے ہم HTML کے <head> یا <body> ٹیگ کے اندر <script> کا ٹیگ لگاتے ہیں اور اس کے بعد JavaScript کا کوڈ لکھتے ہیں۔

```
<script> JavaScript Code Here </script>
```

شروع کرتے ہیں نیچے درج JavaScript کا ہم پہلا کوڈ لکھتے ہیں۔ پھر ہم دیکھیں گے۔

```
<script>
```

```
document.write ("this is my first Java Script Code");
```

```
</script>
```

جیسا کہ آپ کو پہلے ہی بتا چکا ہوں کہ <script> کے ٹیگ کو آپ <head> یا <body> میں کہیں بھی استعمال کر سکتے ہیں تو آپ <script> کے ٹیگ کو کہیں بھی لکھیں کوئی فرق نہیں پڑتا۔ اب آگے بڑھتے ہیں اور کوڈ کو سمجھتے ہیں۔ ہم نے جو <script> کے ٹیگ کے بعد کوڈ لکھا ہے وہ "document.write" ہے اور جیسا کہ نام سے ہی ظاہر ہے اور اگر آپ سمجھ رہے ہیں کہ یہ کوئی ڈوکیومنٹ لکھے گا تو آپ بالکل صحیح ہیں۔ کیوں کہ ہم JavaScript کے کوڈ کو اپنے پیج میں پرنٹ کرنے کے لئے اسی کوڈ کا استعمال کرتے ہیں۔ چونکہ JavaScript کا سارا کام بیک گراؤنڈ میں ہوتا ہے اس لئے ہمیں اس کام کو اپنے پیج پر دکھانے کے لئے document.write کا کوڈ استعمال کرتے ہیں۔ یہ بنیادی طور پر DOM کا حصہ ہے جس کے بارے میں ہم آگے تفصیل سے پڑھیں گے۔

اب اس کوڈ کو تفصیل سے دیکھتے ہیں۔ یہ کوڈ پانچ حصوں میں ہے۔ ہم نے سب سے پہلے "document" لکھا پھر ایک (dot) "." لگایا یہ ڈوٹ دراصل کون کیڈنیشن کے لئے استعمال ہوتا ہے یعنی JavaScript کے دو کوڈز کو آپس میں ملانے کے

لئے اور پھر اس کے بعد ہم نے "write" لکھا اور دو عدد "()" روٹنڈ بیکٹ (round brackets) لگائے اور اس کے بعد ہم نے ایک عدد semi-colon ":" لگایا ہے۔ semi-colon کا کام دو statements کو الگ الگ کرنا ہوگا ہے۔ semi-colon کو آگے تفصیل سے سمجھیں گے۔

اب شاید آپ کو یہ سب کچھ سمجھنے میں مشکل ہو رہے ہو اسی لئے کوڈ دوبارہ لکھتا ہوں دیکھیں۔

document.write ();

اس کوڈ کے بعد جو چیز ہمیں پرنٹ کرنا ہوتی ہے وہ ہم بریکٹس "()" کے اندر لکھتے ہیں۔ اور لکھنے کے لئے بھی کچھ ضروری باتیں یاد رکھیں کہ اگر ہم کوئی لفظ یا الفاظ پرنٹ کروا رہے ہیں تو وہ آپ " " ڈبل یا ' ' سنگل کوٹس کے اندر لکھتے ہیں لیکن اگر وہ کوئی نمبر ہے تو ہم اس کو ڈائریکٹ لکھیں گے یعنی اس میں کسی بھی قسم کے کوٹس کی ضرورت نہیں ہے اس کے علاوہ اگر وہ کوئی ویریبل (variable) ہے جس کے بارے میں بعد پڑھیں گے تو وہ بھی ہم ڈائریکٹ لکھیں گے۔

اب اس کوڈ کو دیکھیں اس کوڈ اور اس کا رزلٹ دیکھیں۔

document.write ("this is my first Java Script Code");

document.write (555);

پہلے کوڈ میں ہم نے ڈبل کوٹس " " کو لگایا ہے کیونکہ وہ ایک string یعنی کے letters ہیں جبکہ دوسرے والے کو ہم نے ڈائریکٹ لکھا ہے کیونکہ وہ number ہے اس لئے ہمیں اس میں کوٹس کی ضرورت نہیں۔

ان دونوں کا رزلٹ اس طرح ہوگا۔

this is my first Java Script Code

555

Methods

JavaScript کو ویب پیج کے اندر ایڈ کرنے کے دو طریقہ کار ہیں۔ ایک ہے انٹرئل (Internal) اور دوسرا ہے ایکسٹرنل (External)۔ انٹرئل کوڈ کو ہم HTML کے کسی بھی ویب پیج کے اندر `<script></script>` ٹیگ کے اندر لکھتے ہیں۔ جیسا کہ ہم پڑھ چکے ہیں۔ Script کے ٹیگ کو ہم HTML کے پیج کے کسی بھی حصہ یعنی Head اور Body میں کسی بھی ٹیگ کے اندر لکھ سکتے ہیں۔ جبکہ ایکسٹرنل کے لئے ہم JavaScript کے لئے ایک نئی فائل بناتے ہیں جس کی ہم ایکسٹینشن ".js" رکھتے ہیں مثال کے طور پر میں JavaScript کے نام سے ایک فائل بناتا ہوں۔ اور اس کو سیو کرتا ہوں .js کی ایکسٹینشن کے ساتھ دیکھیں۔

JavaScript.js

اور اس فائل کو ہم HTML کے کسی بھی ویب پیج کے اندر Script کا ٹیگ لگا کر اور Src کا آرڈریوٹ استعمال کر کے ویب پیج کے اندر اس کا Path دیتے ہیں۔ جیسے کہ اگر میں نے javascript.js کی فائل بنائی اور اس فائل کو میں نے HTML کے فولڈر میں ہی رکھا ہے تو میں اس فائل کو javascript.js کا صرف نام لکھ کر Path دوں گا اور اس کو Script کے ٹیگ کے اندر Src کو فائل کا نام دے کر سیو کروں گا تو وہ فائل میرے ویب پیج کے اندر ایڈ ہو جائے گی۔

```
<script src="javascript.js"></script>
```

Comments

دنیا کی ہر پروگرامنگ زبان کو منٹس کے بغیر ادھوری ہے۔ یہ وہ کوڈز ہوتے ہیں جو ویب پیج پر نظر نہیں آتے لیکن ایک ڈیولپر کے لئے ہوتے ہیں اور اس میں لکھی گئی ہدایت کی بنا پر ایک ڈیولپر کوڈ کو آسانی سے سمجھ سکتا ہے۔ آپ لوگوں نے HTML اور CSS پہلے ہی کو منٹس پڑھے ہو گئے اسی طرح JavaScript کے اندر بھی کو منٹس ہوتے ہیں۔ کو منٹس کا استعمال ہم اپنے کسی کوڈ کی نشاندہی یا کے

کرتے ہیں۔

JavaScript کے اندر سنگل لائن کو منٹ لکھنے کے لئے آپ کو دو فارو ڈسلیش کے بعد ہدایت لکھنی ہوگی اس کا طریقہ یہ ہے اور آخر میں کچھ بھی نہیں ہوتا۔

```
// This is my first javascript code
```

جبکہ آپ کو ایک سے زیادہ لائن کوڈ لکھنے کے لئے دو فارو ڈسلیش کے درمیان میں دو سٹیرک کے درمیان ہدایت لکھنی ہوتی ہیں۔ اس کا طریقہ یہ ہے۔

```
/* This is my first javascript code
and I am learning Javascript comment class with
Muhammad Danish Irshad */
```

Data Types

(1) **String**: JavaScript کے اندر جتنے بھی Letters کو sentence کے طور پر پرنٹ کرنا ہو تو وہ string کہلاتے ہیں اور ان کو سنگل ' ' اور ڈبل کوٹس " " کے اندر لکھا جاتا ہے۔ لیکن اگر وہ یہاں letters ہیں تو وہ ڈائریکٹ پرنٹ ہوں گے۔

مثال :- `document.write("this is the example of string");`

(2) **Integer**: JavaScript کے اندر جتنے بھی نمبرز ہوتے ہیں ان کی ڈیٹا ٹائپ Integer ہوتی ہیں۔ بشرطیکہ وہ string کے طور پر پرنٹ نہ ہو رہے ہوں۔

`document.write (00923142018202);`

مثال :-

(۲) **Boolean** : اس کے اندر صرف Yes اور No کی مانند ہوتی ہیں۔ Boolean کی ڈیٹا ٹائپ کا استعمال ہم آگے کلاسوں میں کریں گے۔ درج ذیل مثال میں ہم نے `document.write` کے فنکشن کے اندر Boolean کے فنکشن لکھا اور Boolean کے فنکشن سے کہا کہ اگر 9 چھوٹا ہو 10 سے تو ہماری ویلیو صحیح ہو جائے۔ اس کا جواب true میں آئے گا۔

`document.write (Boolean (10>9));`

مثال :-

(۴) **Array** : یہ ویریبلز کا مجموعہ ہوتا ہے۔ ویریبلز کے بارے میں ہم آگے پڑھیں گے۔ درج ذیل مثال میں ہم نے ایک Fruits کے نام سے ویریبل بنایا جس کو ہم نے تین ویلیو دیں Mango, Apple, Orange اور ان سب کو `document.write` کے فنکشن کے اندر پرنٹ کر دیا اور جو آپ fruits کے ویریبل کے ساتھ اسکوئر بریکٹس دیکھ رہے ہیں ان میں ہم کو پوزیشن بتانی ہوتی ہے کہ ہمیں کون سے نمبر کی ویلیو درکار ہے۔ چونکہ ہمارے پاس تین ویلیوز ہیں اسی لئے ہم نے نمبر 0 لکھا ہے اور کہا کہ ہمیں 0 نمبر کی ویلیو پر پرنٹ کرنی ہے۔ نوٹ: JavaScript کے اندر پہلا نمبر 0 سے شروع ہوتا ہے۔

`var Fruits = ["Mango", "Apple", "Orange"];`

مثال :-

`document.write(Fruits[0]);`

(۵) **Object** : یہ فنکشنز کا مجموعہ ہوتے ہیں۔ فنکشنز کے بارے میں بھی ہم آگے پڑھیں گے۔ درج ذیل مثال میں ہم نے Array کی ہی طرح ایک car کے نام سے ویریبل بنایا جس کو ہم نے تین ویلیو دیں type:"Ferrari", Made:2015, color:"Red" اور ان سب کو `document.write` کے فنکشن کے اندر پرنٹ کر دیا اور جو آپ car کے ویریبل کے ساتھ کرلی بریکٹس دیکھ رہے ہیں صرف یہی فرق ہوتا ہے objects اور Array کے ویریبل بنانے میں لیکن اس میں آپ نے دکھا کہ میں نے type:"Ferrari" لکھا ہے اس کا مطلب ہے کہ میں نے دو ویلیو دی ہیں ایک type اور type کو میں نے Ferrari کی ویلیو دی ہے یعنی کہ میں نے ایک ویریبل بنایا جس کو ایک ویلیو دی type اور اس type کو بھی Ferrari کی ویلیو دی۔ یہ والا حصہ کافی حد تک CSS جیسا ہے اس لئے آپ کو اس کو سمجھنے میں مشکل نہیں ہونی چاہئے۔ لیکن object میں ہم جب پرنٹ کرتے ہیں تو ہم کو car کا

ویرتبل پرنٹ کرنا ہے اپنے ویب پیج پر لیکن یہاں ہم car کے ویرتبل کے ساتھ اس ویرتبل کی ویلو بھی پرنٹ کریں گے اور چونکہ ہمارے پاس تین ویلوں ہیں اس لئے ہم اس میں " . " لگا کر وہ ویلوں کے جو ہمیں چاہئے جیسے ہم کو بتانا ہے کہ car کی type کیا ہے تو ہم نے car کے ساتھ " . " لگا کر type لکھ دیا اور پرنٹ کروا دیا تو جب ویب پیج پر پرنٹ ہوگا تو ہمیں رزلٹ Ferrari نظر آئے گا۔ object اور Array میں صرف اتنا فرق ہے کہ ہم کو یہاں ویلو دینا ہوتی ہے جبکہ Array میں صرف پوزیشن دینا ہوتی ہے۔ اس کے متعلق ہم آگے بھی پڑھیں گے۔

مثال :- `var car = {type:"Ferrari", Made:2015, color:"Red"};`

`document.write(car.type);`

Variables

اس کلاس میں ہم ویرتبل (Variable) کے بارے میں پڑھیں گے۔ variable کو سمجھنا بہت آسان ہے۔ آپ لوگوں نے الجبرا (Algebra) تو اسکول میں پڑھائی ہوگا۔ یعنی

$$x + y = z$$

بالکل اسی طرح variables بھی ہوتے ہیں۔

مثلاً: x کی ویلو ہے 5 اور y کی ویلو ہے 5 تو ہمیں ان کو جمع کر کے z کی ویلو معلوم کرنی ہے۔

$$x = 5;$$

$$y = 5;$$

$$z = x + y;$$

$$z = 10$$

اب اصل الجبرائیں semi-colon نہیں لگتے لیکن چونکہ ہم JavaScript لکھ رہے ہیں اسی لئے ہم نے semi-colon لگایا ہے۔ جیسا کہ پہلے بتا چکا ہوں کہ semi-colon Statements کو الگ الگ کرتا ہے تاکہ وہ دونوں ایک نہ بن جائیں اور کوئی Error نہ آجائے۔

مثال کے طور پر ہم نے ایک کوڈ لکھا:-

```
document.write ("hello world")
```

```
document.write ("I am writing a JavaScript Code");
```

آپ دیکھ سکتے ہیں کہ ہم نے صرف ایک کوڈ میں semi-colon لگایا ہے تو JavaScript غلط ہوگی اور یہ ایک ہی کوڈ سمجھا جائے گا جس کا مطلب ہمیں پریشانی ہونے والی ہے۔ اور بعض جگہ یہ Error بھی کھڑا کر دے گا۔ اسی لئے ہم ہر کوڈ یا statement کے بعد semi-colon لازمی لگاتے ہیں۔

اب دوبارہ variables کی طرف آتے ہیں۔ variables کی دو نائپ ہوتی ہیں ایک ہے Local variables اور دوسری ہے Global variables ان دونوں کا فرق آگے جا کر واضح کریں گے۔ لیکن پہلے variables دراصل ایک خالی بکس کی طرح ہوتے ہیں جس میں ترتیب سے کوئی چیز رکھی جاتی ہے اور بعد میں ضرورت کے تحت اس میں سے چیز نکال کر استعمال کی جاتی ہے۔ اسی طرح JavaScript میں آپ دیکھیں گے کہ variables میں ہم JavaScript کا کوڈ Save کرتے ہیں اور بعد میں استعمال بھی کرتے ہیں۔ جیسا کہ آپ نے دیکھا تھا میں نے Variable کے لئے الجبرا کی مثال دی تھی جس میں ہم نے x اور y کو 5، 5 کی ویلوی تھی اور z کی ویلوی ان دونوں کو جمع کر کے معلوم کی تھی۔ اب x اور y کی ویلو کچھ بھی ہو سکتی تھی اور ان دونوں کو جمع کر کے ہم z کے علاوہ a, b, c, d کسی کی بھی ویلو معلوم کر سکتے ہیں۔

یقیناً ابھی سمجھنا مشکل ہو رہا ہوگا۔ لیکن جیسے جیسے ہم آگے بڑھیں گے آپ کو variable کا concept زیادہ واضح ہوتا جائے گا۔ اب دیکھتے ہیں کہ variable لکھتے کس طرح ہیں اور ان کو لکھنے کی شرائط کیا ہیں۔

شرائط:

(1) کوئی بھی ویریا بل انگریزی کے حرف letter سے شروع ہوگا۔

- (2) ویربل میں نمبروں کا استعمال ہو سکتا ہے لیکن پہلا حرف انگریزی کا letter ہی ہوگا۔
- (3) ویربل میں آپٹیشل کریکٹر صرف " - " ڈیش اور " _ " انڈر اسکور ہی استعمال ہو سکتے ہیں۔
- (4) آپ ویربل میں سپیس (Space) کا استعمال نہیں کر سکتے۔
- (5) ویربل کا نام Reserve-Keyword میں نہیں رکھا جاسکتا۔ مثال: *if, this* وغیرہ۔

اگر ان میں سے کسی بھی شرائط کو نہ مانا گیا تو Error ہوگا اور آپ کا کام نہیں ہوگا۔ اور آپ مشکل میں پڑ جائیں گے۔

Reserve-Keyword کے متعلق ہم بعد میں پڑھیں گے۔ JavaScript میں variables کو لکھنے کے دو طریقے ہیں۔ ایک طریقہ یہ ہے کہ ہم سب سے پہلے Var کا ایک Keyword لکھیں پھر اسکے بعد variable کا نام اور ایک عدد " = " یہ equal کا سائن ہے لیکن اس کا مطلب یہ equal نہیں بلکہ JavaScript میں ہم اسے Assign Operator کہتے ہیں۔ جن کی تفصیل کے لئے ایک عہدہ کلاس ہوگی۔ اور equal کے سائن کے بعد ہم variable کو ویلو دیں گے۔ جو کہ کوئی نمبر، letter یا کوئی JavaScript کا کوڈ ہوگا اس کے بعد ہم " ; " semi-colon کا سائن لگائیں گے تاکہ statement پوری ہو سکے یعنی کے variable پورا ہو گیا۔

var my_var_name = "write the value of variable" ;

اس میں آپ دیکھ سکتے ہیں کہ میں نے ایک var keyword استعمال کیا ہے۔ آپ دیکھ سکتے ہیں میں نے keyword کو وضع کرنے کے لئے اسکو **bold** اور *italic* کر دیا ہے جس سے آپ کو اسکی وضع پہچان ہو رہی ہوگی۔ پھر اس کے بعد میں نے variable کو ایک نام دیا جو کہ my_var_name کے نام سے ہے۔ میں اسکو کوئی بھی نام دے سکتا تھا۔ اس کے بعد ایک عدد equal کا سائن لگایا جو variable کو ویلو Assign کرنے کے کام آتا ہے۔ پھر اس کے بعد میں نے اس کو ایک string ویلو دی یعنی کے ایک text کو ویلو کے طور پر دیا کہ وہ چونکہ وہ letters ہیں اسی لئے وہ ڈبل کوٹس کے اندر لکھیں ہیں اور آخر میں ایک semi-colon لگایا تاکہ یہ variable پورا ہو سکے۔ اس لائن کو دوبارہ لکھنے کا مقصد صرف اتنا تھا کہ جن لوگوں کو اوپر سمجھنے میں مشکل پیش آرہی ہو تو ان کو دوبارہ سمجھنے میں آسانی ہو۔

اس کو ایک اور انداز میں سمجھا سکتا ہوں کہ ہم نے ایک box لیا جس نام ہم نے my_var_name رکھ دیا پھر اس کے اندر write the value of variable کو لکھ دیا اور semi-colon لگا کر بند کر دیا۔ اب جب بھی مجھے write the value of

variable کی ضرورت ہوگی میں وہ box کھولوں گا اور اس لائن کو پڑھ لوں گا یا اگر کسی کو دینا ہو تو دے دوں گا۔

اب دوسرے طریقے کی طرف آتے ہیں تو وہ بھی بالکل اسی کی طرح سے صرف اس میں ہم keyword کا استعمال نہیں کرتے کیونکہ JavaScript ایک سمجھ دار زبان کے اسی لئے ہم keyword کے بنا بھی variable بنا سکتے ہیں اور ایک اور وجہ بھی ہے Var keyword لکھنے کی اور دہرے سمجھنے کے لئے آپکو فنکشن کی طرف جانا ہوگا جو ہماری اگلی کلاس ہوگی۔ لیکن نئے سیکھنے والوں کے لئے یہ ضروری ہے کہ وہ keyword کا استعمال کریں تاکہ انکو کسی مشکل کا سامنا نہ کرنے پڑے۔ JavaScript میں variables کا استعمال کیسے کرتے ہیں یہ دیکھ لیں۔ اب میں بنا keyword کے variables بناؤں گا۔ اور الجبرا کو ہی استعمال کروں گا۔

میرے پاس ایک x ہے جس کی ویلو ہے 5 اور ایک y ہے جس کی ویلو ہے 6 اور اب مجھے ان کو جمع کر کے z کی ویلو معلوم کرنی ہے۔

x = 5;

y = 6;

z = x + y;

z = 11

اب جیسا کہ آپ دیکھ سکتے ہیں ہم نے سب سے پہلے ایک x نام کا ویربل بنایا اور اس کو 5 کی ویلو دے دی پھر اس کے بعد ایک y نام کا ویربل بنایا اور اس کو 6 کی ویلو دے دی پھر ایک z کے نام سے ویربل بنایا جس کے اندہ ہم نے x کو y سے جمع کر دیا۔ اب ہو گا یہ کہ z کے اندر x اور y کی ویلو جمع ہو جائے گی اور ہمیں جواب z کے ویربل کے اندر ملے گا۔ لیکن یہ ہمارے ویب پیج کے اندر نظر نہیں آئے گا۔ کیونکہ یہ سارا کام بیک گراؤنڈ میں ہوا ہے اور اسکو ویب پیج کے اندر show کرانے کے لئے ہمیں document.write کا استعمال کرنا ہوگا۔ تو اس کو ہم کچھ اس طرح سے اپنے ویب پیج کے اندر show کروائیں گے۔

<script>

x = 5;

y = 6;

z = x + y;

document.write ("this is the answer of z = " + z);

</script>

اب اس کے اندر آپ دیکھ سکتے ہیں کہ document.write کے اندر ہم نے کچھ string یعنی letters پرنٹ کئے ہیں اور z کا ویریبل بھی ساتھ پرنٹ کیا ہے۔ z ایک letter ہونے کے باوجود بھی بنا سٹنگل یا ڈبل کوٹس کے اسی لئے لکھا گیا ہے کیونکہ وہ ایک ویریبل ہے۔ اگر ہم اس کو بھی کوٹس کے ساتھ لکھتے تو وہ ایک string ویریبل بن جاتی اور ویریبل نہ بنتا۔ اور آپ دیکھ سکتے ہیں کہ میں نے string اور ویریبل کے درمیان ایک + کا سائن بھی موجود ہے تو اس کو ہم دو ویریبل کو جوئن کرنے کے لئے استعمال کرتے ہیں۔ جیسے JavaScript کے اسٹینڈرڈس کو جوئن کرنے کے لئے ہم (dot) . کا استعمال کرتے ہیں۔

اس کا رزلٹ کچھ اس طرح ہوگا۔

this is the answer of z = 11

Functions

کچھ فنکشن user define ہوتے ہیں جیسے کہ ہم اس کلاس میں پڑھیں گے اور کچھ JavaScript کے اپنے ہوتے ہیں جیسے document.write () ; وغیرہ ان کو ہمیں Call نہیں کرنا پڑتا جب تک کہ وہ user define فنکشن کے اندر استعمال نہ ہوں۔ فنکشن بنانے کے لئے ہمیں function نام کے ایک keyword کا استعمال کرنا ہوگا اور پھر اس فنکشن کا نام پھر دو رائونڈ بریکٹس اور دو کرلی بریکٹس کا استعمال کرنا ہوگا۔ کچھ اس طرح سے دیکھیں

function function-name () { } آپ کو سمجھانے کے لئے میں سب سے پہلے JavaScript کے فنکشنز function اور document.write کا استعمال کر کے سمجھاؤں گا۔ سب سے پہلے آپ ایک فنکشن بنائیں myfunction کے نام سے، نیچے دیکھیں۔

function myfunction () { }

اب ہم کرنی بریکٹ کے اندر `document.write ()` کے فنکشن کا استعمال کریں گے اور اس کے اندر ایک string دیں۔
"Hello world" تحریر کریں۔ نیچے دیکھیں۔

```
function myfunction ( ) {  
    document.write ("Hello world");  
}
```

جی ہمارا فنکشن پورا ہوا لیکن یہ ہمارے پیج پر ابھی بھی نظر نہیں آئے گا، حالانکہ ہم نے `document.write ()` کا استعمال بھی کیا ہے لیکن اس کے باوجود بھی یہ فنکشن ہمیں نظر نہیں آئے گا۔ اسکی وجہ یہ ہے کہ کسی بھی user define فنکشن کو بنانے کے بعد اس کو پیج کیا اندر call کرنا پڑتا ہے۔ جس کے لئے بھی ہمارے پاس کچھ طریقہ ہوتے ہیں جن میں سے ایک آپ فنکشن کا نام فنکشن کے باہر تحریر کر دیں تو وہ ویب پیج کے اندر نظر آنے لگتا ہے۔ دیکھیں کیسے:-

```
function myfunction ( ) {  
    document.write ("Hello world");  
};  
myfunction ();
```

جس کے بعد اس کا رزلٹ کچھ اس طرح ہوگا۔

Hello world

امید کے آپ کو ابھی تک فنکشن کے بارے میں کافی حد تک سمجھ آئی ہوگی۔ لیکن اگر نہیں سمجھ میں آیا تو ہم ایک اور فنکشن بناتے ہیں۔ ہم اس فنکشن کو بٹن کی مدد سے اپنے پیج کے اندر show کریں گے۔ اس فنکشن کا کام یہ ہوگا کہ HTML کے پیج کے اندر ہمارا کچھ ٹیکسٹ لکھا ہوا ہوگا اور وہیں ایک بٹن ہوگا جب ہمارا بٹن اس بٹن پر کلک کرے گا تو ہمارے پیج کا ٹیکسٹ خود بخود تبدیل ہو کر وہاں آپ کے کمپیوٹر کا وقت اور تاریخ تحریر کر دے گا۔ اور پھر اس کے اندر `document.getElementById` کے نام سے ایک کوڈ لکھا جو

HTML کے ٹیگ سے id سے کوئی بھی چیز اٹھا لیتا ہے۔ اس کام کے لئے ہم نے ایک فنکشن بنایا جس کا نام ہم نے

display_date رکھا ہے اور پھر ہم اس پر JavaScript کی مدد سے کام کر سکتے ہیں۔

document.getElementById() دراصل JavaScript کا ایک فنکشن ہے۔

document.getElementById() سے ہم HTML کے کسی بھی ٹیگ میں موجود مواد کو اس کی Id کے Attribute کی

مدد سے تبدیل کر سکتے ہیں۔ اس کو لکھنے کے بعد ہمیں Id کے رائٹڈ بریکٹس کے اندر Id کا نام لکھنا ہے جو ہم HTML کے کسی ٹیگ کے

اندر استعمال کریں گے۔ اس کے بارے میں یاد رکھیں کہ جیسا یہ لکھا ہوا ہے آپ کو بالکل ویسا ہی لکھنا ہے اگر ایک لفظ بھی غلط ہوا یا کوئی بھی

حرف چھوٹا یا بڑا ہوا تو سارا سارا کوڈ بیکار ہو جائے گا اور کچھ بھی کام نہیں کرے گا۔ جو حرف چھوٹے ہیں وہ چھوٹے لکھے جائیں گے اور جو

بڑے ہیں وہ بڑے حرف میں تحریر ہونگے۔ document.getElementById() کو تحریر کرنے کے بعد ہم ایک (dot) .

لگائیں گے کیونکہ ہمیں ایک اور چیز بھی اس کے ساتھ ملانی ہے اور اسی کے ساتھ لکھنا ہے۔ جو کہ innerHTML کا ایک کوڈ ہے

innerHTML کے استعمال سے ہم HTML کے کسی بھی مواد کو تبدیل کر سکتے ہیں۔ اور ہم یہی کریں گے۔ نیچے دیکھیں کیسے لکھا

ہے:-

```
function display_date( ){
```

```
document.getElementById ("changedate") . innerHTML = Date ();
```

```
};
```

ہم نے innerHTML کو Date () نام کے ایک فنکشن کی ویلو Assign کی ہے Date() فنکشن کے بارے میں ہم آگے

بات کریں گے۔ لیکن اس سے پہلے ہم HTML کے کا ایک <p> کا ٹیگ استعمال کریں گے۔ اور اس کو ایک id دیں گے

changedate کے نام سے اور پھر اس کے بعد ہم ایک button بنائیں گے جس میں ایک Attribute استعمال کریں گے

onclick کے نام کا جس کے اندر ہم فنکشن کا نام لکھیں گے۔ نیچے دیکھیں کس طرح لکھا ہوا ہے۔

```
<p id="changedate">Welcome to this site</p>
```

```
<button onclick="display_date ();">Click Here</button>
```

اب اس کارڈ لٹ دیکھیں۔

بٹن پر کلک کرنے سے پہلے

Welcome to this site

Click Here

بٹن پر کلک کرنے کے بعد

Sat Sep 12 2015 03:12:05 GMT-0500 (Pakistan Standard Time)

Click Here

String Method

یہاں ہم String کے بارے methods کے بارے میں پڑھیں گے۔ String دراصل حرف یا الفاظ ہوتے ہیں۔ جیسے ABCD وغیرہ۔ اس کو سمجھنے کے لئے نیچے درج مثال دیکھیں۔

```
var carName1 = "SERCI 360";
```

```
var carName2 = 'Serci 360';
```

```
document.write (carName1+"<br>" + carName2)
```

اس کے اندر ہم نے دو ویریبل بنائے ہیں ایک carName1 اور carName2 کے نام سے اور دونوں کو ایک ہی نام دیا ہے بس ایک کو چھوٹے اور دوسرے کو بڑے حرف میں اور ان دونوں کو document.write کے اندر پرنٹ کر دیا۔ اس کا رزلٹ دیکھیں۔

SERCI 360

Serci 360

اس میں دونوں بلکل ویسے ہی پرنٹ ہوئے ہیں جیسے دیکھ رہے ہیں۔

اب اگلی مثال میں ہم txt.length کا استعمال کریں گے اس کے استعمال سے ہمیں کسی بھی string ویلو کے حرفوں کے بارے میں پتا چلتا ہے۔

```
<p id="txlh"></p>
```

یہاں ہم نے سب سے پہلے ایک p کا ٹیگ بنایا اور اس کے اندر id بنائی txlh کے نام سے آپ id کو کوئی بھی نام دے سکتے ہیں میں نے صرف مثال کے لئے یہ نام دیا ہے۔

```
<script>
```

```
var txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
```

```
document.getElementById("txlh").innerHTML = txt.length;
```

```
</script>
```

اس کے بعد ہم نے script کے ٹیگ کے اندر اپنی اسکرپٹ لکھی۔ ہم نے پہلے txt کے نام سے ایک ویریبل بنایا جس میں ہم نے A سے لے کر پوری Z تک تمام کے تمام انگریزی کے حرف گنی لکھ لئے۔ اگلی لائن میں ہم نے document.getElementById کا استعمال کیا ہے ہم نے اس کے بارے میں آگے پڑھیں گے۔ آپ اسے فنکشن بھی کہہ سکتے ہیں یہ اونجکت بھی کہہ سکتے ہیں۔ ہم نے document.getElementById کے اندر اپنی p کے ٹیگ والی id جو کہ txlh تھی لکھ دی اور اس کے بعد innerHTML کا استعمال کیا اور اس کو ویلو دے دی txt.length کی اس سے اب جو ہمارا p کا ٹیگ ہے وہ خالی ہے لیکن اگر اس میں کچھ مواد ہوتا تو وہ تبدیل ہو جاتا innerHTML کی وجہ سے اور نتیجہ وہی آتا جو ابھی آیا ہے۔ خالی ہونے کے باوجود بھی یہ اپنا کام کرے گا اور کچھ نہ ہونے کی باوجود یہ کچھ بھی نہیں کو تبدیل کر کے txt.length کو p کے ٹیگ میں تبدیل کر دے گا۔

اس کا رزلٹ یہ ہوگا۔

اس اگلی مثال میں ہم چھوٹے حرف کو بڑے حرف میں تبدیل کریں گے۔ کوڈ دیکھیں۔

```
<button onclick="myFunction()">click to Upper Case</button>
```

```
<p id="upcase">Hello World!</p>
```

اس کے لئے سب سے پہلے ہم نے ایک بٹن بنایا اور اس میں onclick کے event کا استعمال کیا اس کے بارے میں ہم بعد میں تفصیل سے پڑھیں گے۔ onclick کے اندر ہم نے myFunction کے نام سے جو Function بنایا ہے اس کا نام لکھا ہے اور پھر Details کے لئے Click to Upper Case لکھا ہے۔ اس کے بعد ہم نے ایک p کا ٹیگ بنایا اور اس کو id دی upcase کی اور اس ٹیگ کے اندر ہم نے Hello World! لکھا ہے۔

```
<script>
```

```
function myFunction() {
```

```
var text = document.getElementById("upcase").innerHTML;
```

```
document.getElementById("upcase").innerHTML = text.toUpperCase();
```

```
}
```

```
</script>
```

اس کے بعد ہم نے script کے ٹیگ کے اندر ایک فنکشن بنایا myFunction کے نام سے اور اس فنکشن کے اندر ایک ویریبیل بنایا text کے نام سے اور اس ویریبیل کو ویلو اسائن کی document.getElementById کی اور اس کے اندر upcase کی id لکھ دی اور اسے concatenate کر دیا innerHTML کے ساتھ اس سے اب ہماری ساری ویلو text کے ویریبیل میں محفوظ ہو گئی۔ اس کے بعد ہم نے ایک document.getElementById کا پھر استعمال کیا اور اس کے اندر بھی ہم نے دوبارہ upcase کی id لکھ دی اور پھر سے ہم نے اسے innerHTML کے ساتھ concatenate کر دیا۔ اور

innerHTML کو ویلو کے لئے ہم نے جو text کا ویریبل بنایا تھا اس کے ساتھ toUpperCase() کے فنکشن کو concatenate کر کے یہ ویلو دے دی۔

اب جب ہمارا ایوزر بٹن پر کلک کرے گا تو ہمارے سارے حرف بڑے ہو جائیں گے۔

اس کا رزلٹ دیکھیں۔

کلک سے پہلے	کلک کے بعد میں
<div>click to Upper Case</div> <p>Hello World!</p>	<div>click to Upper Case</div> <p>HELLO WORLD!</p>

یہی کام ہم نے اس کوڈ میں بھی کیا ہے لیکن اس بار ہم نے toUpperCase() کے فنکشن کی جگہ toLowerCase() کے فنکشن کا استعمال کیا ہے۔ کوڈ دیکھیں۔

```
<button onclick="myFunction()">click to Lower Case</button>
```

```
<p id="lowcase">Hello World!</p>
```

اس کوڈ میں بھی آپ دیکھ سکتے ہیں وہی سب ہے لیکن id جو پچھلے کوڈ میں upcase تھی یہاں پر lowcase ہے اور نام سے ہی ظاہر ہے یہ حرف کو چھوٹا کر دے گا۔

```
<script>
```

```
function myFunction() {
```

```
var text = document.getElementById("lowcase").innerHTML;
```

```
document.getElementById("lowcase").innerHTML = text.toLowerCase();
```

```
}
```

```
</script>
```

اس کارزلٹ یہ ہوگا۔ اس رزلٹ میں آپ دیکھ سکتے ہیں کہ H اور W دونوں بڑے ہیں جو بعد میں کلک کے بعد چھوٹے ہو گئے ہیں۔

کلک کے بعد میں	کلک سے پہلے
<div>click to Lower Case</div> <p>hello world!</p>	<div>click to Lower Case</div> <p>Hello World!</p>

اس اگلی مثال میں ہم لفظ کی پوزیشن نکالیں گے۔

```
<p id="cpos"></p>
```

اس میں آپ دیکھ سکتے ہیں کہ ہم نے پہلے ایک p کا ٹیگ بنایا اور اس کے اندر ایک id بنائی اور اس id کا نام ہم نے cpos دے دیا۔

```
<script>
```

```
var str = "HELLO WORLD";
```

```
document.getElementById("cpos").innerHTML = str.charAt(2);
```

```
</script>
```

اس کوڈ کے اندر ہم نے str کے نام سے ایک ویریبل بنایا اور اس کو ویلو دے دی HELLO WORLD کی بڑے حرف سے پریشان مت ہوں یہاں اس سے کوئی فرق نہیں پڑتا۔ یہاں ہمیں ابھی صرف اس کی پوزیشن معلوم کرنی ہے۔ اگلی لائن میں ہم نے ایک document.getElementById کے فنکشن کا استعمال کیا اور اس کو جو id ہم نے بنائی تھی اس کا نام دے دیا cpos اور پھر اس کو innerHTML کے ساتھ concadinate کر دیا اور innerHTML کو ویلو دے دی str.charAt(2) کی اس میں ہم نے str جو ہمارا ویریبل ہے اسے charAt() کے ساتھ concadinate کیا ہے اور جو 2 لکھا ہے اس سے ہمیں ہمارے لفظ کی پوزیشن ملے گی۔

اس کارزلٹ یہ ہوگا۔

L

آپ دیکھ سکتے ہیں کہ ہم نے 2 لکھا تھا تو جواب L آیا جب کہ HELLO WORLD میں L تیسرا لفظ ہے۔ لیکن JavaScript کے اندر گنتی 0 سے شروع ہوتی ہے اسی لئے L کی پوزیشن 2 نمبر پر ہے اور 0 نمبر پر H ہوگا۔

اب اگلی مثال میں ہم نے حرف کا کوڈ معلوم کریں گے اور اس کام کے لئے ہم: charCodeAt() کا استعمال کریں گے۔ کوڈ دیکھیں۔

```
<p id="chcode"></p>
```

یہاں بھی ہم نے پہلے ایک p کا ٹیگ بنایا اور اس میں ایک id کو نام دیا chcode اور اب ہمارا سارا کام اسی id پر ہوگا۔

```
<script>
```

```
var str = "HELLO WORLD";
```

```
document.getElementById("chcode").innerHTML = str.charCodeAt(0);
```

```
</script>
```

اس کوڈ کے اندر ہم نے پہلے ہی کی طرح سارا کام کیا ہے بس id کا نام تبدیل ہے اور: charAt() کی جگہ charCodeAt() کا استعمال کیا ہے۔ جس سے ہمیں اس کا کریکٹر کوڈ مل جائے گا۔ یہاں آپ کو بتاتا چلوں کہ دنیا میں جتنے بھی لفظ یا symbols ہیں سب کا کوئی نہ کوئی کریکٹر کوڈ ہے۔ اور یہ اسی سے پہچانے جاتے ہیں۔ اس جگہ ہم نے 0 نمبر کی پوزیشن کا کوڈ مانگا ہے جو کہ حرف H ہے۔

اس کا رزلٹ یہ ہوگا۔

حرف H کا کوڈ 72 ہے۔

اب پڑھتے ہیں escape کریکٹر کے بارے میں اس کام کے لئے ہم ہمیشہ \ (backslash) کا استعمال کرتے ہیں۔

کوڈ	تفصیل
\'	یہ کسی اسٹرنگ لائن میں سنگل کوٹس کے لئے استعمال ہوتا ہے۔
\"	یہ کسی اسٹرنگ لائن میں ڈبل کوٹس کے لئے استعمال ہوتا ہے۔
\\	یہ کسی اسٹرنگ لائن میں سلش کے لئے استعمال ہوتا ہے۔
\n	یہ کسی اسٹرنگ لائن میں نئی لائن کے لئے استعمال ہوتا ہے۔
\t	یہ کسی اسٹرنگ لائن میں tab کے لئے استعمال ہوتا ہے۔
\b	یہ بیک اسپیس کے لئے استعمال ہوتا ہے۔
\f	یہ فارم فیڈ کے لئے استعمال ہوتا ہے۔

Numbers Method

یہاں ہم Numbers کے بارے میں پڑھیں گے۔ جیسا کہ نام سے ہی ظاہر ہے یہ نمبرز ہیں۔ جیسے 12345 وغیرہ۔
اس کو مزید سمجھنے کے لئے کوڈ دیکھیں۔

```
var x = 34.00;
```

```
var y = 34;
```

```
document.write( x + "<br>" + y );
```

اس کا رزلٹ یہ ہوگا۔ اس میں decimal ویلو بھی بغیر decimal کے پرنٹ ہوتی ہے۔ لیکن اگر ہمیں decimal ویلو چاہئے تو اس

کا الگ طریقہ ہے۔

34

34

نمبر کے ویریل کو ہم کسی بھی اوپریٹر کے ساتھ لکھ سکتے ہیں لیکن اگر ہم کسی نمبر کو کسی اسٹرنگ ویلو کے ساتھ جمع کریں گے تو جواب NaN آئے گا۔ کوڈ دیکھیں۔

```
document.write( 100 / "Apple" );
```

اس کا رزلٹ یہ ہوگا۔

NaN

Math Objects

یہاں ہم Math کے انجیکشنس کے بارے میں پڑھیں گے۔ اس کوڈ میں ہم سب سے پہلے Math.random() کا استعمال کر رہے ہیں اس کے استعمال سے ہر بار جب آپ پیج کو ریفریش کریں گے آپ کو ایک نیا نمبر ملے گا۔ کوڈ دیکھیں۔

```
document.write(Math.random());
```

اس کا رزلٹ ایسا ہوگا۔ ہر بار یہ ویلو الگ ہوگی۔

0.22974724345840514

Math.random(); کو ہم ایک اور طریقے سے استعمال کر سکتے ہیں اس کا استعمال عام طور پر گیم کے لئے کیا جاتا ہے۔ پہلے کوڈ دیکھیں۔

```
document.write(Math.random() / 11);
```

اب اس کوڈ سے ہمارا رینڈم نمبر اب صرف 11 سے ہی تقسیم ہوگا اور جواب میں ہمیں ہر بار صرف 0 سے لے کر 10 تک کے نمبرز ملیں گے۔ اس کا رزلٹ ایسا ہوگا۔ ہر بار یہ ویلو الگ ہوگی۔

6

یہاں ہم min اور max کے بارے میں پڑھیں گے اس کو لکھنے کے لئے ہم Math.min(); اور Math.max(); لکھتے ہیں۔

```
document.write(Math.min(0, 150, 30, 20, -8, -200));
```

```
document.write(Math.max(0, 150, 30, 20, -8, -200));
```

یہاں ہم نے Math.min(); اور Math.max(); کو پانچ پانچ ویلو دی ہیں ان میں سے ایک سب سے چھوٹی ویلو کو Math.min اور دوسری سب سے بڑی ویلو کو Math.max حاصل کرتا ہے اور document.write کے اندر اسے پرنٹ کرتا ہے۔

اس کا رزلٹ ایسا ہوگا۔

-200

150

یہاں ہم Math.round(); کا استعمال کریں گے۔ Math.round(); سب سے قریبی ویلو کو حاصل کرتا ہے۔ اس کو مزید سمجھنے

کے لئے یہ کوڈ دیکھیں۔

```
document.write(Math.round(4.4));
```

```
document.write(Math.round(4.5));
```

اس کوڈ میں ہم نے دو `Math.round` کے انجکشنس کا استعمال کیا ہے ایک کی ویلو 4.4 ہے اور دوسری کی ویلو 4.5 ہے۔ اور جیسا کہ سب جانتے ہیں کہ `Math` کے Rule کے مطابق اگر پونٹ یا ڈوٹ کے بعد کی ویلو 5 سے کم ہو تو پہلی ویلو ہی گنی جاتی ہے جب کہ اگر وہ ویلو 5 کے برابر یا زیادہ ہو تو پہلی ویلو میں ایک کا اضافہ کر دیا جاتا ہے۔

اس کا رزلٹ ایسا ہوگا۔

4
5

math کے کچھ مزید objects یہ ہیں۔

کود	تفصیل
<code>cos(x)</code>	اس سے ہم cosine معلوم کر سکتے ہیں۔
<code>tan(x)</code>	اس سے ہم کسی Angle کا tangent معلوم کر سکتے ہیں۔
<code>sin(x)</code>	اس سے ہم sine حاصل کر سکتے ہیں۔
<code>log(x)</code>	اس سے ہم logarithm حاصل کر سکتے ہیں۔
<code>sqrt(x)</code>	اس سے ہم سکوائر روٹ کا حاصل کر سکتے ہیں۔
<code>pow(x,y)</code>	اس سے ہم x کے اوپر y کا پاور حاصل کر سکتے ہیں۔

Events

یہاں ہم Events کے بارے میں پڑھیں گے۔ Events یہ ہے کہ کلائنٹ کے کچھ کرنے پر جو رد عمل آئے وہ Events ہوتے ہیں۔ مثلاً اگر کوئی ماؤس کا کوئی بٹن دبائے اور یا کسی پیراگراف کے اوپر لے جائے یا اپنے کی بورڈ سے کوئی بٹن دبائے تو اس پیراگراف کا رنگ تبدیل ہو جائے یا اس میں کچھ الفاظ واضح ہو جائیں تو وہ Events کہلاتے ہیں۔ اور وہ functions جو Events پر کام کرتے ہیں Event handling کہلاتے ہیں۔ درج ذیل میں کچھ بنیادی Event واضح کر رہا ہوں جس سے آپ کو انہیں سمجھنے میں آسانی ہوگی۔

Event Handling	جو Events پر فارم کرے
onLoad	جب پیج لوڈ ہو
onKeyDown	جب کی بورڈ کی کوئی کی (key) دبائی جائے
onKeyPress	جب کی بورڈ کی کوئی کی (key) مسلسل دبائے رکھی جائے
onKeyUp	جب کی بورڈ کی کوئی کی (key) دبائی رکھی ہو وہ اوپر واپس آئے
onClick	جب ماؤس سے click کیا جائے
onDbClick	جب ماؤس سے double click کیا جائے
onMouseDown	جب ماؤس کا بٹن دبایا جائے
onMouseUp	جب ماؤس کا دبایا ہوا بٹن اٹھ جائے
onMouseEnter	جب ماؤس کسی object پر لایا جائے
onMouseLeave	جب ماؤس کسی object سے ہٹایا جائے
onMouseOver	جب ماؤس کو کسی object کے اوپر لایا یا کسی object کے اوپر لایا جائے
onMouseMove	جب ماؤس کو کسی object کے اوپر حرکت دی جائے یا گھمایا جائے
onMouseOut	جب ماؤس کسی object سے ہٹایا جائے
onSelect	جب یوزر کوئی چیز مثلاً Text وغیرہ سلیکٹ کرے
onSubmit	جب یوزر submit کے بٹن پر کلک کرے یا فارم submit کرے

اب ایک کوڈ دیکھیں

<h2 onclick="this.style.color= 'blue';">

Click Text to Turn Blue

</h2>

اس کوڈ میں ہم نے ایک onclick Attribute استعمال کیا ہے۔ یہ HTML کا جاوا اسکرپٹ کے لئے ہی بنا Attribute ہے اور ہمارے events میں سے بھی ایک ہے۔ اس کے اندر ہم نے this.style.color کا استعمال کیا ہے اور اس کو ویلڈے دی blue اس کا استعمال ہم تب کرتے ہیں جب ہمیں JavaScript کے اندر کوئی CSS کا اسٹائل دینا ہو۔ اس کوڈ کے تین حصہ ہیں۔ پہلا ہے this اس سے ہم اس کوڈ کو بتاتے ہیں کہ جو کام کرنا ہے وہ یہیں پر کرنا ہے۔ دوسرا style اس سے ہم کوئی بھی CSS کا اسٹائل لگا سکتے ہیں مثلاً style.position، style.color، style.fontSize وغیرہ۔ اور آخر میں ہم نے جو color لکھا ہے اس سے ہمیں پتا چل جاتا ہے کہ یہ اسٹائل کمر کے لئے ہے اور اس اسٹائل کو ہم نے ویلڈے دی blue کی جس سے اب اس کوڈ کی وجہ سے جب بھی کوئی بھی یوزر تحریر پر کلک کرے گا تو وہ نیلے رنگ کی ہو جائے گی۔ یہ محض سمجھانے کے لئے تھا اگر مشکل ہو رہی ہے تو اگلی مثال دیکھیں۔

<div onmouseover="this.style.color= 'blue';

this.style.fontSize="20px;">

Click Text to Turn Blue and increse font size

</div>

اس کوڈ میں ہم نے onmouseover کے event کا استعمال کیا ہے۔ اس کوڈ سے جب کوئی یوزر تحریر پر کلک کرے گا تو اس کا سائز 20px اور کمر نیلا ہو جائے گا۔

اب یہی کام ہم دو تصویروں کے ساتھ کرتے ہیں اس کام کے لئے ہم مندرجہ ذیل دو تصویروں کا استعمال کریں گے۔



```

```

اس کوڈ میں this.src سے تصویر کا سورس تبدیل ہو جائے گا۔ جب کوئی یوزر تصویر پر اپنا ماؤس لے کر جائے گا تو تصویر تبدیل ہو جائے گی اور بند بلب کی جگہ روشن بلب کی تصویر آجائے گی اور ایسا لگے گا جیسے ماؤس لے کر جانے سے بلب جل گیا ہو اور جب ماؤس دوبارہ ہٹایا جائے گا تو پھر پہلے والی تصویر دوبارہ آجائے گی اور بلب پھر بند ہو جائے گا۔

```
<h2 onmouseover="this.innerText='mouse out to change text again';"
```

```
onmouseout="this.innerText='Thank you';">
```

```
mouse over me to change text
```

```
/h2>
```

اس مثال میں ہم نے this.innerText کا استعمال کیا ہے اس سے ہماری تحریر تبدیل ہو جائے گی۔

Assignment Operator

یہ آسائنمنٹ اوپریٹر ہے۔

Assign

=	Assign
---	--------

یہ کسی بھی ویریبل کو ویلویں کے کام آتا ہے یہ equal نہیں اس کام کے لئے ہم اسے دوبار استعمال کرتے ہیں۔ مثال کے طور پر ہم ایک ویریبل کو ایک ویلویں دینا چاہتے ہیں تو اس کو ویلویں کے لئے ہم "=" کے سائن کا استعمال کریں گے تاکہ اس ویریبل کو ویلویں سکے۔ یہاں ہم نے ایک ویریبل بنایا myvar_1 کے نام سے اور اس کو ایک String ویلویں دی۔

Var myvar_1 = "the data of variable is string";

Arithmetic Operators

یہ آریٹھمٹک اوپریٹرز ہیں۔

Addition

+	Add
---	-----

یہ دو ویریبلوں یا دو نمبرز کو جمع کرنے کے لئے استعمال ہوتا ہے۔ مثال کے طور پر ہم نے دو ویریبل بنائے۔ ایک x کے نام سے جس کو ہم نے 5 کی ویلویں دی اور ایک y کے نام سے جس کو ہم نے 5 کی ویلویں دی اور دونوں کو ہم نے document.write(); کے فنکشن کے اندر جمع کا اوپریٹر لگا کر جمع کر دیا۔

```
var x = 5;
var y = 5;
document.write(x + y);
```

اس کا رزلٹ یہ ہوگا۔

10

یہاں + کا سائن ان دونوں کو ایک ساتھ ملا کر نہیں لکھنے کے لئے استعمال نہیں ہو رہا کیونکہ یہ دونوں ویریبلز ہیں اور دونوں کو جمع ہونا ہے۔ اگر یہ سٹرنگ ہوتے اور سنگل یا ڈبل کوئٹس کے اندر لکھے ہوتے تو آپ دونوں x اور y الگ الگ لکھے ہوئے دیکھتے۔

Subtract

-	Subtract
---	----------

یہ دو ویریبلز یا دو نمبرز کو منفی کرنے کے لئے استعمال ہوتا ہے۔ مثال کے طور پر ہم نے دو ویریبل بنائے۔ ایک x کے نام سے جس کو ہم نے 5 کی ویلو دے دی اور ایک y کے نام سے جس کو ہم نے 15 کی ویلو دے دی اور دونوں کو ہم نے document.write(); کے انکشن کے اندر مائنس کا اوپریٹر لگا کر گھٹا کر دیا۔

```
var x = 5;
var y = 15;
document.write(x - y);
```

اس کا رزلٹ یہ ہوگا۔

5

Multiplay

*	Staric / Multiplay
---	--------------------

یہ اسٹیرک کا سائن ہے۔ یہ دو ویریبلز یا دو نمبرز کو ضرب (x) کرنے کے لئے استعمال ہوتا ہے۔ مثال کے طور پر ہم نے دو ویریبل بنائے۔ ایک x کے نام سے جس کو ہم نے 5 کی ویلو دے دی اور ایک y کے نام سے جس کو ہم نے 5 کی ویلو دے دی اور دونوں کو ہم نے document.write() کے فنکشن کے اندر اسٹیرک کا سائن لگا کر ضرب کر دیا۔

var x = 5;

var y = 5;

document.write(x * y);

اس کا رزلٹ یہ ہوگا۔

25

Divide

/	Slash / Divide
---	----------------

یہ سلش کا سائن ہے جو کہ تقسیم (÷) کے لئے استعمال ہوتا ہے۔ یہ دو ویریبلز یا دو نمبرز کو تقسیم کرنے کے لئے استعمال ہوتا ہے۔ مثال کے طور پر ہم نے دو ویریبل بنائے۔ ایک x کے نام سے جس کو ہم نے 15 کی ویلو دے دی اور ایک y کے نام سے جس کو ہم نے 3 کی

ویلو دے دی اور دونوں کو ہم نے `document.write()` کے فنکشن کے اندر سلیش کاسائن لگا کر تقسیم کر دیا۔

```
var x = 15;
var y = 3;
document.write(x / y);
```

اس کا رزلٹ یہ ہوگا۔

5

Modulas

%	Percentage / Modulas
---	----------------------

اس کو سمجھنا حوڑا مشکل ہو گا یہ موڈیولس یا پرنسپل کا سائن کہلاتا ہے۔ یہ بھی Divide کی طرح کام کرتا ہے لیکن Divide ہونے کے بعد جو رقم بچ جاتی ہے وہ رقم یہ دیتا ہے۔ جیسے ہم نے پہلے Divide کیا تو 3 کو پانچ مرتبہ جمع کرنے سے 15 بنتا ہے یعنی 15 کو 3 سے ضرب کرنے پر جواب 5 ہوتا ہے تو وہ Divide کا جواب تھا۔ لیکن موڈیولس میں ایسا نہیں ہوتا یہاں 3 اگر 5 سے ضرب ہو جاتا ہے تو جواب 0 آئے گا۔ لیکن اگر 3 کو 17 سے ضرب کریں تو 15 تک ضرب کرنے کے بعد جب ضرب نہیں ہو سکے گا تو جو بچ جائے گا وہ جواب ہو گا یعنی 2 جواب ہو گا۔ اس کو سمجھنے کے لئے دو مثالیں دیکھیں۔

مثال 1 میں ہم نے دو ویرئبل بنائے۔ ایک x کے نام سے جس کو ہم نے 15 کی ویلو دے دی اور ایک y کے نام سے جس کو ہم نے 3 کی ویلو دے دی اور دونوں کو ہم نے `document.write()` کے فنکشن کے اندر جمع کا اوپر لگا کر جمع کر دیا۔

```
var x = 15;
var y = 3;
document.write(x % y);
```

اس کا رزلٹ یہ ہوگا۔

0

مثال 2 میں ہم نے دو ویرتبل بنائے۔ ایک x کے نام سے جس کو ہم نے 17 کی ویلو دے دی اور ایک y کے نام سے جس کو ہم نے 3 کی ویلو دے دی اور دونوں کو ہم نے document.write() کے فنکشن کے اندر جمع کا اوپر پریٹر لگا کر جمع کر دیا۔

```
var x = 17;
```

```
var y = 3;
```

```
document.write(x % y);
```

اس کا رزلٹ یہ ہوگا۔

2

یہاں آپ دیکھ سکتے ہیں 3 جب تک ضرب ہو سکتا تھا ہو گیا یعنی کے 15 تک لیکن چونکہ 3 کہ پھاڑے میں 17 نہیں ہوتا اس لئے وہ وہیں رک گیا اور جواب ہمارے پاس 2 آ گیا۔ اگر آپ کو سمجھنے میں مشکل لگے تو بار بار پڑھیں اور پرنٹس کریں۔ یقیناً جلد سمجھ آ جائے گا۔

Increment ++

++

Increment ++

یہ سائن دیکھ کر آپ کو لگ رہا ہوگا کہ یہ دوبار جمع کرے گا لیکن ایسا نہیں ہے۔ یہ دوبار جمع نہیں کرتا بلکہ یہ کسی بھی نمبر کے اندر 1 کا اضافہ

کر دیتا ہے۔ مثال کے طور پر ہم نے ایک ویریبیل بنایا۔ x کے نام سے جس کو ہم نے 5 کی ویلو دے دی اور اس کو لکھنے سے پہلے ++ کا سائن لگا دیا۔ اس کو ہم نے document.write() کے فنکشن کے اندر پرنٹ کر دیا۔ یاد رکھیں ++ کا سائن آپ کو پہلے لگانا ہے بعد میں ورنہ ویلو نہیں دے گا۔

```
var x = 5;
document.write(++x);
```

اس کا رزلٹ یہ ہوگا۔

6

Decresement --

--	Decresement --
----	----------------

++ والے سائن کی طرح یہ بھی دوبارہ مانس نہیں کرتا بلکہ یہ کسی بھی نمبر کے اندر 1 کی کمی کر دیتا ہے۔ مثال کے طور پر ہم نے ایک ویریبیل بنایا۔ x کے نام سے جس کو ہم نے 5 کی ویلو دے دی اور اس کو لکھنے سے پہلے -- کا سائن لگا دیا۔ اس کو ہم نے document.write() کے فنکشن کے اندر پرنٹ کر دیا۔ دوبارہ بتا رہا ہوں کہ یاد رکھیں -- کا سائن آپ کو پہلے لگانا ہے بعد میں ورنہ جو ویلو آپ چاہتے ہیں وہ آپ کو نہیں ملے گی۔

```
var x = 5;
document.write(--x);
```

اس کا رزلٹ یہ ہوگا۔

4

Comparison Operators

یہ کمپوزن اوپریٹرز ہیں۔

Equal

==	Equal
-----------	--------------

ہم جیسا کہ پہلے پڑھ چکے ہیں ڈبل equal کا مطلب ہے equal تو اس کا استعمال ہم تب کرتے ہیں جب ہمیں یہ دیکھنا ہو کہ ہمارے پاس جو دو ویلیوز ہیں وہ ایک برابر ہیں یا نہیں۔ مثال کے لئے ہم دو ویریبل بناتے ہیں ایک x کے نام سے اور اس کو 5 کی ویلیو دے دیتے ہیں اور دوسرا y کے نام سے اور اس کو بھی ہم 5 کی ویلیو دے دیتے ہیں اور ایک if کے فنکشن کا استعمال کر کے دیکھتے ہیں کہ یہ دونوں برابر ہیں یا نہیں۔ اگر یہ دونوں برابر ہوں گے تو ایک میسج آئے گا کہ یہ دونوں برابر ہیں ورنہ کوئی میسج نہیں آئے گا۔ ہم if کے بارے میں آگے تفصیل سے پڑھیں گے۔

```
var x = 5;
```

```
var y = 5;
```

```
if ( x == y) {
```

```
document.write("The two variables are equal");
```

```
}
```

```
4
```

اس کا رزلٹ یہ ہوگا۔

The two variables are equal

Equal to Type

===	Equal to Type
-----	---------------

اس میں آپ دیکھ رہے ہیں کہ اس میں تین equal کے سائن ہیں اس کو ہم تب استعمال کرتے ہیں جب ہمیں معلوم کرنا ہو کہ اگر ہمارے پاس جو ڈیٹا ہے اس کی ویلو اور ڈیٹا ٹائپ دونوں برابر ہیں یہ نہیں۔ اس کو سمجھنے کے لئے ہم دو ویریبل بناتے ہیں۔ ایک x کے نام سے جس کو ہم نے 5 کی ویلو دیتے ہیں اور ایک y کے نام سے جس کو ہم نے 5 کی ویلو دیتے ہیں اور پھر ہم ایک فنکشن استعمال کرتے ہیں if کا اور دیکھتے ہیں کہ آیا اگر دونوں کی ویلو اور ڈیٹا ٹائپ ایک ہیں تو ہمیں ایک میسج اپنے پیج پر دکھاؤ۔

```
var x = 5;
```

```
var y = 5;
```

```
if (x === y) {
```

```
document.write("both variables have same DataType and Value");
```

```
}
```

اس کا رزلٹ یہ ہوگا۔

both variables have same DataType and Value

Not

!	Not
---	-----

جیسا کہ نام سے ہی ظاہر ہے یہ Not یعنی نہیں کے لئے استعمال ہوتا ہے۔ مثال کے طور پر اگر ہم چاہیں کہ ہمارے پاس دو ویریبلز ہوں اور اگر دونوں کی ویلو ایک نہ ہو تو ہمیں ایک پیغام ویب پیج میں نظر آئے تو اس کام کے لئے ہم Not کا استعمال کرتے ہیں۔ اس کو سمجھنے

کے لئے ہم دو ویریبلز بناتے ہیں۔ ایک کو ہم x کا نام دیتے ہیں جس کو ہم نے 5 کی ویلو دیتے ہیں اور دوسرے کو ہم y کا نام دیتے ہیں جس کو ہم 6 کی ویلو دیتے ہیں اور اس کے بعد ہم ایک if کا فنکشن بناتے ہیں اس فنکشن کے اندر ہم چاہتے ہیں کہ اگر ہمارے دونوں ویریبلز کی ویلو مختلف ہو تو ہمیں ایک پیغام نظر آئے۔ اس کام کے لئے ہم equal کے سائن کے بھی استعمال کریں گے۔

```
var x = 5;
var y = 6;
if (x !== y) {
    document.write("variable x and y has not same value");
}
```

اس کا رزلٹ یہ ہوگا۔

variable x and y has not same value

اسی طرح ہم چاہتے ہیں کہ اگر دونوں کی ٹائپ بھی ایک نہ ہو تو بھی ہمیں ایک پیغام نظر آئے تو ہم اس کام کے لئے تین بار equal کا سائن لگائیں گے اور y کو ویلو ہم string میں دیں گے۔

```
var x = 5;
var y = "Hello";
if (x !== y) {
    document.write("variable x and y has not same Data type");
}
```

اس کا رزلٹ یہ ہوگا۔

variable x and y has not same Data type

Greater Than

<	Greater Than
---	--------------

یہ HTML کے ٹیگ کو اشارت کرنے کا پہلا اسائن ہوتا ہے۔ اس کا استعمال ہم کسی بھی دو ویلیوز میں بڑے یا چھوٹے کی تفریق کے لئے استعمال ہوتا ہے۔ مثال سے سمجھتے ہیں ہم نے دو ویریبیل بنائے۔ ایک کو x کا نام دیا اور اس کو ہم نے 5 کی ویلیو دے دی اور ایک کو y کا نام دیا اور اس کو ہم نے 6 کی ویلیو دے دی اور اس کے بعد ہم نے ایک if کنڈیشن بنایا۔ اس کنڈیشن کے اندر ہم چاہتے ہیں کہ اگر ہمارے دونوں ویریبلز میں سے y کی ویلیو بڑی ہو تو ہمیں ایک پیغام نظر آئے۔

```
var x = 5;
```

```
var y = 6;
```

```
if (x < y) {
```

```
document.write("value of the y is greater than x");
```

```
}
```

اس کا رزلٹ یہ ہوگا۔

value of the y is greater than x

Less Than

>	Less Than
---	-----------

یہ HTML کے ٹیگ کو ختم کرنے کا دوسرا اسائن ہوتا ہے۔ اس کا استعمال ہم کسی دو ویلیوز میں بڑے یا چھوٹے کی تفریق کے لئے

کرتے ہیں۔ مثال سے سمجھیں ہم نے دو ویربل بنائے۔ ایک کو x نام دیا اور اس کو ہم نے 5 کی ویلو دے دی اور ایک کو y نام دیا اور اس کو ہم نے 4 کی ویلو دے دی اور اس کے بعد ہم نے ایک if کنڈیشن بنایا۔ اس کنڈیشن کے اندر ہم چاہتے ہیں کہ اگر ہمارے دونوں ویربلز میں سے y کی ویلو چھوٹی ہو تو ہمیں ایک پیغام نظر آئے۔

```
var x = 5;
```

```
var y = 4;
```

```
if (x > y) {
```

```
document.write("value of the y is Less then x");
```

```
}
```

اس کا رزلٹ یہ ہوگا۔

value of the y is Less then x

Greater Then or Equal

<=	Greater then or Equal
----	-----------------------

جیسا کہ نام سے ہی ظاہر ہے اگر ایک ویربل کی ویلو دوسرے ویربل سے زیادہ ہو یا پھر کم از کم اس کے برابر ہو تو رزلٹ ہمیں دے گا۔ اس کام کے لئے بھی ہم if کنڈیشن کا استعمال کریں گے۔

مثال نمبر 1:-

```
var x = 5;
```

```
var y = 5;
```

```
if (x <= y) {
```

```
document.write("Value of the variable y is Greater then equal to x");
```

```
}
```

اس کا رزلٹ یہ ہوگا۔

Value of the variable y is Less then or equal to x

مثال نمبر 2:-

```
var x = 5;
```

```
var y = 8;
```

```
if (x <= y) {
```

```
document.write("Value of the variable y is Greater then equal to x");
```

```
}
```

اس کا رزلٹ یہ ہوگا۔

Value of the variable y is Greater then or equal to x

Less Then or Equal

>=

Less Then or Equal

جیسا کہ نام سے ہی ظاہر ہے اگر ایک ویرتبل کی ویلوی دوسرے ویرتبل سے کم ہو یا پھر کم از کم اس کے برابر ہو تو رزلٹ ہمیں دے اور اس

کام کے لئے بھی ہم if کے فنکشن ہی کا استعمال کریں گے۔

مثال نمبر 1:-

```
var x = 5;
var y = 5;
if (x >= y) {
document.write("Value of the variable y is Less then or equal to x");
}
```

اس کا رزلٹ یہ ہوگا۔

Value of the variable y is Less then or equal to x

مثال نمبر 2:-

```
var x = 5;
var y = 8;
if (x >= y) {
document.write("Value of the variable y is Less then or equal to x");
}
```

اس کا رزلٹ یہ ہوگا۔

Value of the variable y is Less then or equal to x

Logical Operators

یہ لو جیکل اوپر پٹرز ہیں۔

Addition

+	Add
---	-----

یہ دو ویریبلس کا جمع کرنے کے لئے استعمال ہوتا ہے۔ مثال کے طور پر ہم نے دو ویریبلس بنائے۔ ایک x کے نام سے جس کو ہم نے 5 کی ویلو دے دی اور ایک y کے نام سے جس کو ہم نے 5 کی ویلو دے دی اور دونوں کو ہم نے document.write(); کے فنکشن کے اندر جمع کا اوپر پٹر لگا کر جمع کر دیا۔

```
var x = 5;
var y = 5;
document.write(x + y);
```

اس کا رزلٹ یہ ہوگا۔

10

if

if کا مطلب ہے اگر مثال کے لئے انگریزی کا جملہ "if you have a bike" اگر آپ کے پاس بائک ہے۔ اس جملے میں کمی نظر آتی ہے جیسے اگر آپ کے پاس بائک ہے تو آپ کیا کریں گے؟ تو اب ہمیں یہ بتانا ہے کہ اگر آپ کے پاس بائک ہے تو آپ کیا کریں۔ JavaScript میں بھی if اسی کام کے لئے استعمال ہوتا ہے۔ if کے فنکشن کو لکھنے کے لئے ہم function ہی کی طرح if

کے keyword کے بعد دو رائنڈ بریکٹس اور دو کرلی بریکٹس لگاتے ہیں۔ رائنڈ بریکٹس کے اندر ہم کنڈیشن لکھتے ہیں اور کرلی بریکٹس کے اندر ہم کوڈ لکھتے ہیں جو کام ہم کو کروانا ہوتا ہے۔

```
if ( ) { }
```

اس کو سمجھنے کے لئے ہم ایک کوڈ لکھتے ہیں۔ سب سے پہلے دو ویربل بنائیں a اور b کے نام سے اور ان کو مختلف ویلیوز دیں۔ پھر if کے کنڈیشن کے اندر ان کو لکھیں کہ آیا اگر a کی ویلیو زیادہ ہو تو ہمیں ایک میسج دیکھا دیں۔ یعنی اگر a کی ویلیو b سے بڑی ہوگی تو ہی ہمیں رزلٹ ملے گا ورنہ ہمیں کچھ نظر نہیں آئے گا۔

```
var a = 12;
```

```
var b = 15;
```

```
if ( a < b ) {
```

```
document.write ("if you have a bike go to bazaar");
```

```
}
```

اس کا رزلٹ یہ ہوگا۔

```
if you have a bike go to bazaar
```

else

else کو سمجھنا بھی آسان ہے۔ else کا مطلب ہے مگر یعنی if اگر اور else مگر اس کو سمجھنے کے لئے ایک انگریزی کا جملہ دیکھیں۔

"if you have a bike go to bazaar else"

اس میں ہم نے یہ تو بتا دیا کہ "اگر آپ کے پاس بانک ہے تو آپ اپنے بازار جائیں ورنہ" اب ہمیں اس کو یہ بتانا ہے کہ ورنہ وہ کیا کرے اگر اس کے پاس بانک نہ ہو تو وہ پھر اس صورت میں کیا کرے۔ تو ہم else کا استعمال اس لئے کرتے ہیں کہ اگر ایسا نہ ہو تو اور کیا کرنا چاہئے۔ else کو لکھنے کا طریقہ بھی بہت آسان ہے آپ else لکھیں اور دو کرنی بریکٹس لکھ دیں کیونکہ اس میں آپ کو کنڈیشن کی ضرورت نہیں۔ ایک سادہ سی مثال دیکھیں اوپر والی ہی مثال کو آگے لے کر چلتے ہیں۔

```
var a = 12;
```

```
var b = 10;
```

```
if ( a < b ) {
```

```
document.write ("if you have a bike go to bazaar");
```

```
}
```

```
else {
```

```
document.write ("if you have a bike go to bazaar else take a bus and go  
to bazaar");
```

```
}
```

اس کارزلٹ یہ ہوگا۔

```
if you have a bike go to bazaar else take  
a bus and go to bazaar
```

آپ یہاں غور کریں کہ میں نے if کے فنکشن میں greater then کا سائن تبدیل نہیں کیا صرف b کی ویلو کم کر دی ہے۔ جس کا مطلب ہے کہ اب b کی ویلو کم ہو گئی ہے۔ جب ہم پہلے کام کرتے تھے تو ہمارے پاس اس غلطی کے بعد کوئی جواب نہیں آتا تھا لیکن اب چونکہ ہم نے else کا آپشن بھی استعمال کیا ہے اسی لئے اب ہمارے پاس ایک اور جواب بھی موجود ہے۔ جو اُنظر آ رہا ہے۔ یعنی اگر a کی ویلو b سے بڑی ہوگی تو ہی ہمیں پہلا والا رزلٹ نظر آئے گا ورنہ دوسرا والا رزلٹ نظر آئے گا۔

else if

یہ بھی if کی ہی طرح ہے اس کا کام بھی یہی ہے لیکن ہم اس کو جب استعمال کرتے ہیں جب ہمیں کسی کام کے متعلق مختلف آپشنز رکھنا ہوں۔ else if کو

مثال کے لئے ہم انگریزی کے اسی جملے کو پھر دہراتے ہیں۔

"if you have a bike go to bazaar else take a bus and go to bazaar or give me back my money"

اب آپ دیکھ سکتے ہیں کہ اس بار جملے میں ہم نے کہا کہ اگر آپ کے پاس بانگ ہے تو آپ بازار جائیں ورنہ آپ بس پکڑیں اور بازار جائیں یا پھر میرے پیسے واپس کر دیں۔

```
var a = 12;
```

```
var b = '10';
```

```
if ( a < b ) {
```

```
document.write ("if you have a bike go to bazaar");
```

```
}
```

```
else if (a > b) {
```

```
document.write ("if you have a bike go to bazaar else take a bus or give me back my money");
```

```
}
```

```
else {document.write ("if you have a bike go to bazaar else take a bus");
```

```
}
```

اس کا رزلٹ یہ ہوگا۔

if you have a bike go to bazaar else take a bus and go to bazaar or give me back my money

Popup Boxes

Popup Boxes تین مختلف اقسام کے ہوتے ہیں۔ جو درج ذیل ہیں۔ یہ بنیادی طور پر Browser Object Model یا BOM کا حصہ ہیں۔ جن کے بارے میں ہم آگے تفصیل سے پڑھیں گے۔

1) Alert Box

alert (): سب سے پہلا Popup Box ہے جس کا استعمال ہم اپنے ویب پیج کے اندر user خوش آمدید کہنے کے لئے یا کسی بھی قسم کی وارنگ دینے کے لئے کرتے ہیں۔ لیکن اس کے استعمال کو زیادہ تر لوگ پسند نہیں کرتے۔ اور کیوں مجباً آپ کو خود ہی پتا چل جائے گی جب آپ اس کو استعمال کریں گے۔ پھر بھی میں آپ کو سکھارہا ہوں۔ دیکھیں کہ alert box کو کس طرح لکھتے ہیں۔

```
alert ("Hello, How are you?");
```

یہ ایک فنکشن ہے۔ ہم اس کو ریٹرنل کے اندر بھی استعمال کر سکتے ہیں۔ اور بعد از ضرورت استعمال میں لا سکتے ہیں۔ اس کا رزلٹ کچھ اس طرح کا ہوگا۔



2) Prompt Box

prompt (): ہمارا دوسرا Popup Box ہے جس کا استعمال ہم اپنے ویب پیج کے اندر user سے کسی بھی قسم کا ڈیٹا حاصل کرنے کے لئے کرتے ہیں۔ اس کے دو پیرامیٹرز ہوتے ہیں۔ ایک جو آپریشن کے طور پر ہوتا ہے اور دوسرا جو کہ user کے ڈیٹا ان پٹ فیلڈ کے اندر ہوتا ہے۔ دیکھیں کہ prompt box کو کس طرح لکھتے ہیں۔


```
prompt ("Please enter you name" , "Danish Irshad");
```

اب جیسے ہی user اس prompt box پر ok کے بٹن پر کلک کرے گا وہ ہمارے ویب پیج کے اندر آ جائے گا۔ ہم اس کو پری میل کے اندر بھی استعمال کر سکتے ہیں۔ دیکھیں کیسے:-

```
var name = prompt ("Please enter your name" , "Danish");  
document.write ("Hello, " + name + " Welcome to our site");
```

اب اس کا رزلٹ کچھ اس طرح ہوگا۔

Hello, Danish Welcome to our site

3) Confirm Box

Confirm (): اس کا استعمال ہم اپنے ویب پیج میں user کے کسی چیز کے متعلق ہاں یا نا میں جواب مانگتے ہیں۔ مثال کے طور پر اگر ہم اپنے user کو کہیں کہ اگر آپ ہماری شرائط کو قبول کرتے ہیں تو Yes پر کلک کریں تو آپ ہماری ویب سائٹ کے اندر داخل ہو سکتے ہیں ورنہ نہیں۔ اس کو اس طرح لکھتے ہیں دیکھیں۔

```
confirm ("Do you agree to our Terms & Services");
```

لیکن ابھی بھی اگر کوئی یوزر No پر کلک کرتا ہے تو وہ بھی ہمارے پیج کے اندر وزٹ کر سکتا ہے کیونکہ ہم نے اس کو کوئی متبادل راستہ کہیں جانے کے لئے نہیں دیا۔ اسی لئے ہمیں اس کے اندر if اور else کے فنکشن کا استعمال کرنا ہوگا۔

```
var x=confirm("Press a button!");  
if (x == true) {
```

```
document.write ("You Pressed Ok!");
```

```
} else {
```

```
document.write ("You Pressed Cancel!");
```

```
}
```

ہم نے یہاں سب سے پہلے ایک x کے نام سے ویریبل بنایا جس کو ہم نے ویلو (); confirm کی دے دی۔ پھر ہم نے ایک if کا فنکشن بنایا جس میں ہم نے if کے فنکشن کو کہا کہ اگر ویریبل x کی ویلو جو کہ confirm box ہم نے دی ہے۔ اگر برابر ہو تو صحیح دیں اور document.write کے ذریعے سے You Pressed ok! پرنٹ کر دیں۔ جس سے اگر کوئی یوزر ok پر کلک کرے گا تو وہ ویلو صحیح ہو جائے گی۔ لیکن اگر کوئی یوزر cancel پر کلک کرتا ہے تو ہم نے ابھی تک نہیں بتایا کہ جاوا اسکرپٹ کو کیا کرنا ہے۔ ہم نے اس کے لئے else کے فنکشن کا استعمال کیا اور جیسا کہ آپ لوگ جانتے ہیں کہ else کا مطلب ہی مگر ہے۔ یعنی کے ہمیں وضاحت کرنے کی ضرورت نہیں۔ اگر کوئی یوزر cancel پر کلک کرتا ہے تو پر else کے فنکشن کی مدد سے document.write کے فنکشن سے You Pressed Cancel! پرنٹ ہوگا۔

اس کا رزلٹ کچھ اس طرح ہوگا۔



for loop

لوپ تین مختلف اقسام کے ہوتے ہیں۔ for loop: ہماری لوپ کی کلاس کا سب سے پہلا لوپ ہے۔ یہ لوپ سب سے زیادہ مشہور اور سب سے زیادہ استعمال ہونے والا لوپ ہے۔ تینوں لوپس میں بہت زیادہ فرق نہیں ہے۔ لیکن فور لوپ کے اندر ہم کسی بھی ویریبل کو اس کے اندر ہی استعمال کرتے ہیں۔ فور لوپ اس طرح لکھا جاتا ہے۔

```
for ( ) { }
```

اس کو تفصیل سے سمجھنے کے لئے آپ نیچے درج کوڈ کو دیکھیں پھر میں آپ کو اس کوڈ کو سمجھاتا ہوں۔

```
for (x=1; x<10; x++) {  
    document.write(x);  
}
```

ہم نے سب سے پہلے ایک فور لوپ بنایا جس کے اندر ہم نے ایک ویریبیل بنایا x کے نام سے اور اس کو ویلو دے دی 1 کی پھر ہم نے x کے ویریبیل کو اس میں ڈیفائن کیا جس کو ہم نے greater than کا سائن لگا کر ایک ویلو دے دی جس کو ہم نے 10 کی ویلو دے دی جس سے اس کوڈ کو ہم نے بتایا کہ اگر x کی ویلو 10 سے چھوٹی ہو تو اس میں ایک کا اضافہ کر دو۔ اب ہمیں کیسے پتا چلے گا کہ یہ اضافہ کہاں ہو رہا ہے اور کس چیز میں ہو رہا ہے تو اس کام کے لئے ہم نے document.write کا استعمال کیا جس میں ہم نے x کو لکھ دیا۔ اب یہ لوپ کام شروع کر چکا ہے جو لگاتار چلتے رہیں گے۔ جب تک کہ کنڈیشن غلط نہیں ہو جاتی۔ یعنی کے جیسے ہی ہمارا 1 نمبر 10 کے برابر ہو جائے گا کنڈیشن غلط ہو جائے گی اور رزلٹ پرنٹ ہونا بند ہو جائے گا اور لوپنگ بند ہو جائے گی۔ اب جب آپ رزلٹ دیکھیں گے تو آپ کو 1 سے لے کر 9 تک نمبرز ملیں گے۔

اس کا رزلٹ یہ ہے۔

```
123456789
```

ہم Increment کے بجائے Decrement کا سائن بھی استعمال کر سکتے ہیں۔ کوڈ دیکھیں۔

```
for (x=20; x>10; x--) {  
    document.write(x);  
}
```


}

ہم نے ایک ویریبیل بنایا x کے نام سے اور اس کو ویلو دے دی 20 کی پھر ہم نے دوبارہ x کا ایک اور ویریبیل بنایا جس کو ہم نے less than کا سائن لگا کر ایک ویلو دے دی جو کہ میں نے 10 دی دے جس سے اس کو ڈکو ہم نے بتایا کہ اگر x کی ویلو 10 سے چھوٹی ہو تو اس میں ایک کی کمی کر دو۔ اب یہ لوپ بار بار چکر لگائے گا۔ جس کے بعد ہم یہ رزلٹ ملے گا۔

987654321

while loop

while loop ہماری لوپ کی کلاس کا دوسرا لوپ ہے۔ یہ بھی فور لوپ کی طرح ہی ہے لیکن اس میں ہم ویریبیل کو باہر ہی ڈیفائن کرتے ہیں۔ اور کنڈیشن کو ہم اسٹیمنٹ کے نیچے کی جانب لکھتے ہیں۔ while loop اس طرح لکھا جاتا ہے۔

while () { }

اس کو تفصیل سے سمجھنے کے لئے آپ نیچے درج کوڈ دیکھیں پھر میں آپ کو اس کوڈ کو سمجھاتا ہوں۔

var x=1;

while (x<10) {

document.write(x);

x++;

}

ہم نے سب سے پہلے ایک ویریبیل بنایا x کے نام سے اور اس کو ویلو دے دی 1 کی پھر ہم نے ایک while loop بنایا اور x کے

ویرتبل کو اس میں ڈیفائن کیا جس کو ہم نے greater than کا سائن لگا کر ایک ویلو دے دی جیسا کہ میں نے یہاں 10 دی ہے جس سے اس کوڈ کو ہم نے بتایا کہ اگر x کی ویلو 10 سے چھوٹی ہو تو اس میں ایک کا اضافہ کر دو۔ اب یہاں بھی ہم وہی کام کریں گے جو ہم نے فورلوپ کے اندر کیا تھا ایک document.write کا استعمال کریں گے جس میں ہم نے x کو لکھ دیا۔ اب ہمیں Increment یا Decrement کرنا ہے جیسا کہ ہم نے فورلوپ کے اندر کیا تھا لیکن یہاں پر یہ اسٹیٹمنٹ کے اندر ہوگا جیسا کہ آپ کوڈ میں دیکھ رہے ہیں۔ میں نے x کے بعد Increment کے سائن document.write کے بعد لکھے ہیں۔ اب یہ لوپ کام شروع کر چکا ہے جو لگاتار چلتے رہیں گے۔ جب تک کہ کنڈیشن غلط نہیں ہو جاتی۔ یعنی کے جیسے ہی ہمارا 1 نمبر 10 کے برابر ہو جائے گا کنڈیشن غلط ہو جائے گی اور زلٹ پرنٹ ہونا بند ہو جائے گا اور لوپنگ بند ہو جائے گی۔ اب جب آپ زلٹ دیکھیں گے تو آپ کو 1 سے لے کر 9 تک نمبرز ملیں گے۔ اس کارزلٹ بھی اس وقت فورلوپ جیسا ہی ہوگا۔

اس کارزلٹ یہ ہوگا۔

123456789

do while loop

do while loop ہماری لوپ کی کا اس کا تیسرا لوپ ہے۔ اس لوپ میں بھی ہم ویرتبل کو باہر ہی ڈیفائن کرتے ہیں لیکن اس میں ہم do کے بعد دو کرلی بریکٹس {} لگاتے ہیں اور اس کے اندر ہم کوڈ لکھتے ہیں۔ پھر ہم while لکھتے ہیں اور پھر ہم دو راؤنڈ بریکٹس () لگاتے ہیں۔ اور اس کے اندر ہم کنڈیشن لکھتے ہیں۔ اس لوپ میں اور for اور while loop میں صرف اتنا فرق ہے کہ اگر کنڈیشن کی ویلو اور ویرتبل کی ویلو ایک جیسی بھی ہوں تو بھی یہ کم از کم ایک دفعہ زلٹ ضرور پرنٹ کرتا ہے لیکن فورلوپ یا while loop کے اندر ایسا نہیں ہوتا اگر ان میں دونوں ویلو ایک ہو جائیں تو زلٹ نہیں آتا لیکن اس میں آ جاتا ہے۔ do while loop اس طرح لکھا جاتا ہے۔

do { } while ()

اس کو تفصیل سے سمجھنے کے لئے آپ نیچے درج کوڈ دیکھیں پھر میں آپ کو اس کوڈ کو سمجھاتا ہوں۔

```

var x=1;

do {

document.write(x);

x++;

} While (x<10)

```

ہم نے سب سے پہلے ایک ویریبل بنایا x کے نام سے اور اس کو ویلو دے دی 1 کی پھر ہم نے do لکھنے کے بعد دو کرنی بریکٹس کا استعمال کیا ہے جس کے اندر ہمیں اپنا کوڈ لکھنا ہے فورلوپ اور while loop کی طرح ہم یہاں بھی document.write کا ہی استعمال کر رہے ہیں کیونکہ ہمیں یہاں بھی اپنے ویریبل x کو پرنٹ کروانا ہے۔ پھر ہم نے اپنے ویریبل x کو Increment کا سائن لگا کر بتا دیا کہ ہمیں اپنے x کے ویریبل میں ایک کا اضافہ چاہیے۔ اب ہم نے while لکھا اور اس کے آگے دو رائونڈ بریکٹس لگا دیے جس میں اب ہم کنڈیشن لکھیں گے اور ہم نے جو کنڈیشن دی ہے اس سے اب یہ لوپ بھی بالکل دوسرے لوپس کی طرح ہی چلے گے۔

اس کا رزلٹ یہ ہوگا۔

123456789

لیکن جیسا کہ میں نے پہلے شروع میں بتایا تھا کہ اگر ویریبل کی ویلو اور کنڈیشن کی ویلو برابر ہو جائے تو بھی یہ رزلٹ ضرور پرنٹ گے گا۔ اب کوڈ دیکھیں۔

```

var x=10;

do {

document.write(x);

x++;

} While (x<10)

```

یہاں آپ دیکھ سکتے ہیں کہ ہم نے ویریل کو بھی 10 کی ویلوی اور کنڈیشن میں بھی بتادیا کہ اگر x کی ویلوی 10 سے کم ہو تو ہی اس میں 1 کا اضافہ کرو۔ لیکن یہاں پر اس میں کوئی فرق نہیں پڑے گا اور ہمارے پاس کم از کم ایک مرتبہ رزلٹ ضرور آئے گا جو کہ آپ نیچے دیکھ سکتے ہیں۔

اس کا رزلٹ یہ ہوگا۔

10

ایسا اس لئے ہوا ہے کہ do پہلے کوڈ کو ایکسی کیوڈ کرتا ہے یعنی پہلے یہ ایک کا اضافہ کر رہا ہے پھر while لوپ کو چلا رہا ہے۔ جس سے ہمیں کم از کم ایک مرتبہ رزلٹ ضرور مل رہا ہے۔ اگر میں ویریل کی ویلوی 10 کی جگہ 9 کر دیتا تو بھی ہمیں صرف ایک ہی رزلٹ ملتا۔

Switch & Case

switch & Case ایک قسم کا لوپ سسٹم ہے۔ جس کے اندر ہم Break کی ممانڈ بھی استعمال کرتے ہیں لوپ کو ختم کرنے کے لئے۔

اسے سمجھنے کے لئے نیچے درج کوڈ دیکھیں۔

```
var name = 'asif';
```

```
switch (name) {
```

```
case 'asif' : document.write ('Hello Asif');
```

```
break;
```

اس کوڈ کے اندر ہم نے سب سے پہلے ایک name کے نام سے ویریل بنایا جس کو ویلوی دی ہم نے asif کی اور پھر ہم نے switch کی ممانڈ استعمال کی جس کے آگے دو رائنڈ بریکٹس کے اندر ہم نے اپنے ویریل کا نام لکھ دیا اور اس کے بعد ہم نے دو کرنی بریکٹس لگائے اور اس کے اندر ہم نے case کا استعمال کیا ہے۔ یہاں ہم نے case کوڈ ایکٹ asif کا نام دے دیا اور اس کے

بعد ہم نے ایک semi-colon لگایا اور پھر ہم نے document.write کا استعمال کیا یہاں میں بتاتا چلوں کہ آپ یہاں document.write کی جگہ if یا alert کا استعمال بھی کر سکتے ہیں جیسے آپ چاہیں۔ آگے چلتے ہیں ہم نے document.write کے اندر Hello Asif لکھا کیونکہ ہم چاہتے ہیں ہے Hello Asif پرنٹ ہو۔ پھر ہم نے اس کے آگے break کی مانند استعمال کیا جس سے ہمارا کوڈ یہاں ختم ہو گیا۔ اب یہاں ہو گا یہ کہ اگر ہمارے ویریل کی ویلو asif ہے تو ہم نے case میں جو ڈائریکٹ asif لکھا تھا وہ ویلو میچ ہو جائے گی اور ہمیں اپنے پیچ پر Hello Asif نظر آئے گا۔

اس کا رزلٹ یہ ہو گا۔

Hello Asif

نوٹ: ہم نے case کے اندر asif کا چھوٹا لکھا ہے جب کہ document.write کے اندر Asif بڑا ہے کیونکہ وہ ہمیں پرنٹ کروانا تھا لیکن Case کے اندر وہ ویریل کی ویلو ہے اگر آپ نے اس میں کچھ گڑبڑ کی وہ رزلٹ نہیں آئے گا۔

اب آگے بڑھتے ہیں اور اس کو اگلے لیول پر لے جاتے ہیں۔ یہاں ہم نے تین case لکھے ہیں اور ہر case کے بعد break کی مانند کو بھی استعمال کیا ہے اور یہ ضروری بھی ہے۔ ورنہ رزلٹ نہیں ملے گا۔

```
name = 'asif';

switch (name) {

    case 'asif' : document.write ('Hello Asif');

        break;

    case 'Danish' : document.write ('Hello Danish');

        break;

    case 'Junaid' : document.write ('Hello junaid');

        break;

}
```


اوپر جو ہم نے تین case لکھے ہیں سب کی ویلو الگ ہیں۔ اب جو بھی ویلو ہمارے ویریبل سے میچ ہوگی ہمیں اس case کا پرنٹ ملے گا۔ لیکن اگر کوئی بھی ویلو میچ نہ ہوئی تو یہ ابھی کوئی رزلٹ نہیں دے گا اور Error کا سبب بنے گا۔ اس کام کے لئے ہم default کی مائنڈ کا استعمال کرتے ہیں جیسے نیچے درج مثال میں ہم نے ویریبل کی ویلو asif سے تبدیل کر کے ahsan کر دی ہے۔ جو کہ ہمارے کسی بھی case سے میچ نہیں ہوتی۔ تو ہم نے default کا استعمال کیا ہے نیچے دیکھیں۔

```
name = 'ahsan';

switch (name) {

    case 'asif' : document.write ('Hello Asif');

        break;

    case 'Danish' : document.write ('Hello Danish');

        break;

    case 'Junaid' : document.write ('Hello junaid');

        break;

    default : document.write ('There is no match');

}
```

اب یہاں ہم نے default کے بعد وہی سب کچھ کیا ہے جو ہم نے case میں کیا تھا اور document.write کے اندر there is no match کو پرنٹ کروا دیا جس کے بعد اگر کوئی ویلو میچ نہ ہو تو رزلٹ یہ ہوگا۔

There is no match

default کے بعد break کی مائنڈ کی کوئی ضرورت نہیں رہتی۔ switch اور case کو مزید سمجھنے کے لئے اگلی مثال کی طرف چلتے ہیں۔ جس سے آپ کو یہ مزید واضح ہو جائے گا۔

```

switch (new Date().getDay()) {
    case 0: document.write("Today is Sunday");
        break;
    case 1: document.write("Today is Monday");
        break;
    case 2: document.write("Today is Tuesday");
        break;
    case 3: document.write("Today is Wednesday");
        break;
    case 4: document.write("Today is Thursday");
        break;
    case 5: document.write("Today is Friday");
        break;
    case 6: document.write("Today is Saturday");
        break;
}

```

اس مثال میں ہم نے date کے فنکشنز کا استعمال کیا ہے۔ new Date() اور getDay() دونوں فنکشنز کا استعمال کیا ہے اور دونوں کو کنکٹڈ کرنے کے لئے ایک dot کا استعمال کیا ہے۔ date کے بارے میں ہم آگے مزید تفصیل سے پڑھیں گے۔ اس date کے اندر ہم نے جس getDay() کا استعمال کیا ہے اس سے ہمارے پاس اب ہفتے کے ساتوں دن آگئے ہیں۔ جو کہ نمبرز کے حساب سے ہوتے ہیں۔ جو 0 سے شروع ہوتے ہیں اور 6 پر ختم ہوتے ہیں۔ ہم نے ہر case کو ہر دن کا ایک نمبر دے دیا جس کے بعد document.write کا استعمال کرتے ہوئے ہم نے ہر case کے ساتھ ایک پیغام پرنٹ کروا دیا۔ یہاں ہم نے ہفتے کے نام ہی پرنٹ کروائے ہیں آپ کچھ بھی پرنٹ کروا سکتے ہیں۔ آج چونکہ ہفتے کا دن ہے اسی لئے آج ہمارے پاس یہ رزلٹ آئے گا۔

Today is Saturday

آپ اسے اپنی ویب سائٹ پر لگا سکتے ہیں جس سے جو بھی یوزر آپ کی ویب سائٹ پر آئے گا۔ اُسے ہر دن ایک نیا پیغام ملے گا۔ آپ اس میں مہینے اور سال کا بھی کر سکتے ہیں۔

Return

جیسا کہ اس کے نام سے ہی ظاہر ہوتا ہے کسی چیز کو واپس کرنے کو Return کہتے ہیں۔ JavaScript میں بھی ہم Return کو کوئی ایک ویلو دیتے ہیں اور وہ ویلو وقت ضرورت ہمیں Return کرتا ہے۔ اس کو ہم ویریل سے کمپیر کر سکتے ہیں۔ اس کو سمجھنے کے لئے ہم ایک فنکشن بناتے ہیں myfunction کے نام سے اور اس میں return کو ایک ویلو دیں گے۔ کوڈ دیکھیں۔

```
function myfunction () {  
    return 'Getting return';  
}
```

یہاں ہم نے ایک فنکشن بنایا myfunction کے نام سے اور اس میں return کا استعمال کیا جس کو ہم نے ایک string ویلو دے دی getting return کی اب ہونا تو یہ چاہیے کہ جب ہم اس فنکشن کو کال کریں تو ریٹرن کی ویلو پرنٹ ہو جائے لیکن ایسا نہیں ہوگا۔ اگر میں صرف myfunction کو کال کروں تو مجھے کوئی رزلٹ نہیں ملے گا۔ مثلاً کوڈ دیکھیں۔

```
function myfunction () {  
    return 'Getting return';  
}  
  
myfunction ();
```

اب اس کوڈ میں ہم نے myfunction کوڈ انٹرکٹ کال کیا لیکن اس کے باوجود ہمارے پاس کوئی رزلٹ نہیں آئے گا۔ اس کام کے لئے ہمیں اپنے فنکشن کو document.write یا پھر alert کے اندر پرنٹ کروانا ہوگا۔ ہم یہاں document.write کا

استعمال کریں گے۔ کوڈ دیکھیں۔

```
function myfunction () {
    return 'Getting return';
}

document.write(myfunction ());
```

اب ہمیں جو رزلٹ ملے گا وہ یہ ہوگا۔

Getting return

Break & Continue

یہ دو شیٹمنٹس ہیں Break اور Continue ان دونوں کو سمجھنے کے لئے ہم ایک فور لوپ کا استعمال کریں۔

```
for (x=1; x<10; x++){
    document.write (x+ '. number line <br>');
}
```

یہاں پر ہم نے ایک فور لوپ بنایا جس میں ایک x کے نام سے ویریبل بنایا جو کہ ہم نے 1 کی ویلو دے دی اور پھر کنڈیشن میں ہم نے بتایا کہ اگر x کی ویلو 10 سے ہم ہو تو 1 کا اضافہ کر دو جس سے اب جتنی بار لوپ چلے گا اس میں ایک کا اضافہ ہو جائے گا۔ پھر ہم نے اس کو پرنٹ کروانے کے لئے document.write کا استعمال کیا۔ یہاں پر آپ دیکھ سکتے ہیں کہ میں نے x کے بعد ایک + کا سائن لگا کر ایک string ویلو دی ہے جس کا مقصد آپ کو آسانی سے سمجھانا ہے۔

اس کا رزلٹ یہ ہوگا۔

1. number line
2. number line
3. number line
4. number line
5. number line
6. number line
7. number line
8. number line
9. number line

اب ہم اسی کوڈ کے اندر if کی اسٹیٹمنٹ کا استعمال کریں گے۔ کوڈ دیکھیں۔

```
for (x=1; x<10; x++){
    if (x == 3) {break;}
    document.write (x+ '. number line <br>');
}
```

یہاں پر ہم نے اسی کوڈ کے اندر if کا استعمال کیا ہے جس میں ہم نے بتایا کہ اگر x 3 کے برابر ہو تو برابر ہو تو یہ اسٹیٹمنٹ بریک ہو جائے تو ہمیں جو رزلٹ آئے وہ ملے۔ اب آپ جب کوڈ کو دیکھیں گے تو یہ لوپ 1 اور 2 کے بعد بریک ہو جائے گا اور آگے یہ لوپ چلنا بند ہو جائے گا۔

اس کا رزلٹ یہ ہوگا۔

1. number line

2. number line

یہ تو کام تھا break کا اور ہم چلتے ہیں continue کی طرف یہاں پر بھی ہم if کا ہی استعمال کریں گے۔ ہم نے اس کوڈ کے اندر بھی بتایا کہ اگر x برابر ہو 3 کے تو یہ اسٹیٹمنٹ continue ہی رہے۔

```
for (x=1; x<10; x++){
  if (x == 3) {continue;}
  document.write (x+ '. number line <br>');
}
```

اس کوڈ سے اب جب ہمارا لوپ 3 تک پہنچے گا تو یہ تین نمبر اس کو skip کر رہے گا اور لوپ کو آگے لے کر چل پڑے گا۔

اس کا رزلٹ یہ ہوگا۔

1. number line

2. number line

4. number line

5. number line

6. number line

7. number line

8. number line

9. number line

Type Conversion

اس کے اندر ہم JavaScript کے کسی ایک ڈیٹا ٹائپ کو کسی دوسرے ڈیٹا ٹائپ میں تبدیل کر سکتے ہیں۔

```
<p id="ctype"></p>
```

```
<script>
```

```
document.getElementById("ctype").innerHTML =
```

```
typeof "Danish";
```

```
</script>
```

سب سے پہلے ہم نے ایک p کا ٹیگ بنایا اور اس کو id دی ctype کی اور اس id پر ہی ہم JavaScript کا سارا کام کریں گے۔ اس کوڈ میں ہم نے سب سے پہلے document.getElementById کا فنکشن استعمال کیا اور اس میں id دی ctype کی آپ کوئی بھی رکھ سکتے ہیں اس کے بعد ہم نے innerHTML کا استعمال کیا اس سے ہم HTML کے کسی بھی ٹیگ کے اندر کے مواد کو تبدیل کر سکتے ہیں۔ اس کو ہم نے ویلویڈی typeof کی یہ JavaScript کی ایک مائنڈ ہے جس سے ہم کسی بھی ڈیٹا کی ٹائپ معلوم کر سکتے ہیں۔ ہم نے اسے ویلویڈی Danish جو کہ ایک اسٹرنگ ویلویڈی ہے تو اس کا رزلٹ بھی اسٹرنگ آنا چاہیے۔

اس کا رزلٹ یہ ہے۔

```
string
```

اس کے اندر بھی ہم نے وہی تمام کام کئے ہیں بس اس مرتبہ ہم نے ڈیٹا ٹائپ تبدیل کر دیا ہے اس مرتبہ ہم نے ویلویڈی دی ہے اس طرح اس کا ڈیٹا ٹائپ بھی تبدیل ہو جائے گا۔

```
<p id="ctype"></p>
```

```
<script>
```

```
document.getElementById("ctype").innerHTML =
    typeof 3.14;
</script>
```

اس کا رزلٹ یہ ہے۔

number

اس کے اندر بھی وہی سب کچھ ہے اور اس کو ہم نے ویلوی ہے NAN کی جس سے اس کا ڈیٹا ٹائپ نمبر آئے گا کیونکہ NAN کی ویلوی پر ڈیفالٹ طور پر ڈیٹا ٹائپ نمبر آتا ہے۔

```
<p id="ctype"></p>
<script>
document.getElementById("ctype").innerHTML =
    typeof NaN;
</script>
```

اس کا رزلٹ یہ ہے۔

number

اس کے اندر ہم نے ویلوی boolean کی دی ہے جیسا کہ آپ جانتے ہیں کہ boolean کی ویلوی true یا false ہوتی ہے تو ان میں سے کوئی بھی ویلوی دینے پر ڈیٹا ٹائپ boolean ہی آئے گی۔

```
<p id="ctype"></p>
```


<script>

document.getElementById("ctype").innerHTML =

typeof false;

</script>

اس کا رزلٹ یہ ہے۔

boolean

اس کے اندر ہم نے ایک Array بنایا ہے لیکن چونکہ Array بنیادی طور پر ایک object ہوتا ہے اس لئے اس کا ڈیٹا ٹائپ بھی object ہی ہوگا۔

<p id="ctype"></p>

<script>

document.getElementById("ctype").innerHTML =

typeof [1,2,3,4];

</script>

اس کا رزلٹ یہ ہے۔

object

یہ ایک object ہے اسی لئے اس کا ڈیٹا ٹائپ object ہوگا۔

<p id="ctype"></p>

<script>

document.getElementById("ctype").innerHTML =

typeof {name:'john', age:34};

</script>

اس کا رزلٹ یہ ہے۔

object

اس کے اندر ہم new Date() کا فنکشن دیا ہے جو کہ بنیادی طور پر JavaScript کا ایک object ہے اسی لئے اس کا ڈیٹا ٹائپ object ہوگا۔

<p id="ctype"></p>

<script>

document.getElementById("ctype").innerHTML =

typeof new Date();

</script>

اس کا رزلٹ یہ ہے۔

object

اس کے اندر ہم نے ایک فنکشن بنایا ہے اسی لئے اس کا ڈیٹا ٹائپ function ہوگا۔

<p id="ctype"></p>

<script>

document.getElementById("ctype").innerHTML =

typeof function () {};

</script>

اس کا رزلٹ یہ ہے۔

function

اس کے اندر ہم نے کوئی ویلیو نہیں دی ایک اسٹرنگ ویلیو myCar دی ہے لیکن ڈبل کوٹس کے بغیر جس سے یہ ایک ویریل بن جاتا ہے۔
لیکن چونکہ اس ویریل کو کوئی ویلیو ابھی تک نہیں ملی ہے اس لئے اس کا ڈیٹا ٹائپ undefined ہوگا۔

<p id="ctype"></p>

<script>

document.getElementById("ctype").innerHTML =

typeof myCar;

</script>

اس کا رزلٹ یہ ہے۔

undefined

اس کے اندر ہم نے myCar کی طرح صرف اسٹرنگ ویلیو دی ہے لیکن JavaScript کے اندر ڈیفالٹ طور پر
null ایک کی ورڈ ہے اس لئے اس کا ڈیٹا ٹائپ object ہوگا۔

```
<p id="ctype"></p>
```

```
<script>
```

```
document.getElementById("ctype").innerHTML =
```

```
typeof null;
```

```
</script>
```

اس کارزلٹ یہ ہے۔

object

constructor Property

ہم اس پروپٹی کا استعمال کسی بھی ڈیٹا ٹائپ کا تعمیری فنکشن معلوم کرنے کے لئے کرتے ہیں یعنی کسی فنکشن کو بنانے کے لئے کس طرح کے فنکشن کی مدد لی جاسکتی ہے۔ کوڈ دیکھیں۔

```
<p id="constr"></p>
```

```
<script>
```

```
document.getElementById("constr").innerHTML =
```

```
"john".constructor
```

```
</script>
```

اس کوڈ میں آپ دیکھ سکتے ہیں کہ ہم نے پہلے ہی کی طرح ایک p کا ٹیگ بنایا اور اس کو id دی constr کی اس کے بعد script کے ٹیگ کے اندر JavaScript کا کوڈ لکھا اور اس کوڈ میں ہم نے document.getElementById کے اندر id لکھ دی جو ہم نے پیر گراف کے ٹیگ کے اندر لکھی تھی۔ اس کے بعد ہم نے innerHTML کا استعمال کر کے ایک اسٹرنگ ویلیو کو constructor کی پروپٹی کے ساتھ concadinate کر دیا۔

اس کارزلٹ یہ ہے۔

```
function String() { [native code] }
```

یعنی string کا فنکشن ہوگا اور native code سے مراد یہاں کوڈ ہوگا۔ جو کچھ اس طرح لکھا جائے گا۔

```
String() {}
```

یہ عام فنکشن کی طرح ہی لگ رہا ہے۔ اور کام بھی تقریباً ایسا ہی کرے گا۔ اس کے بارے میں مزید تفصیل سے آگے پڑھیں گے۔
اگلے کوڈ میں ہم نے وہی کام دوبارہ کیا اور اس بار ہم نے جو ویلیو دی ہے وہ ہے نمبر جس سے اب ہمارے پاس فنکشن نمبر کا آئے گا۔

```
<p id="constr"></p>
```

```
<script>
```

```
document.getElementById("constr").innerHTML =
```

```
(3.14).constructor
```

```
</script>
```

اس کارزلٹ یہ ہے۔

```
function Number() { [native code] }
```

اس بار ہم نے ایک boolean ویلیو دی ہے جو کہ true یا false ہوتی ہے۔ ہم نے false لکھا اس کا مطلب یہ boolean ہے تو
جواب بھی اس کا boolean ہوگا۔

```
<p id="constr"></p>
```

<script>

document.getElementById("constr").innerHTML =

false.constructor

</script>

اس کا رزلٹ یہ ہے۔

function Boolean() { [native code] }

یہاں پر ہم نے ایک Array بنایا ہے اور اس کا جواب ایک Array کا فنکشن ہوگا۔

<p id="constr"></p>

<script>

document.getElementById("constr").innerHTML =

[1,2,3,4].constructor

</script>

اس کا رزلٹ یہ ہے۔

function Array() { [native code] }

یہاں ایک object ہے تو جواب میں ہمیں ایک object کا فنکشن ملے گا۔

<p id="constr"></p>

<script>

```
document.getElementById("constr").innerHTML =
    {name:'john', age:34}.constructor
</script>
```

اس کا رزلٹ یہ ہے۔

```
function Object() { [native code] }
```

یہاں پر ہم نے new Date() کا استعمال کیا ہے۔ یہ ہے تو بنیادی طور پر ایک object لیکن یہ ہمیں Date کا فنکشن دے گا کیونکہ Date کا عائدہ فنکشن ہوتا ہے۔

```
<p id="constr"></p>
<script>
document.getElementById("constr").innerHTML =
    new Date().constructor
</script>
```

اس کا رزلٹ یہ ہے۔

```
function Date() { [native code] }
```

اس بار ہم نے چونکہ فنکشن کا ہی استعمال کیا ہے اس لئے جواب میں بھی ہمارے پاس function ہی آئے گا۔

```
<p id="constr"></p>
<script>
```

```
document.getElementById("constr").innerHTML =
    function () {}.constructor
    </script>
```

اس کا رزلٹ یہ ہے۔

```
function Function() { [native code] }
```

Converting Numbers to Strings

یہاں پر ہم نمبرز کو اسٹرنگ ویلو کے اندر تبدیل کریں گے۔ اس کام کے لئے ہم نے ایک p کا ٹیگ کے اندر numstr کی id بنائی۔

```
<p id="numstr"></p>
```

اس کے بعد ہم نے script کے ٹیگ کے اندر ایک ویریبیل بنایا x کے نام سے اور اس کو ایک نمبر ویلو دے دی 123 کی اس کے بعد ہم نے document.getElementById کا استعمال کرتے ہوئے پیراگراف کے ٹیگ کی id جو ہم نے numstr لکھی تھی دے دی اس کے بعد ہم نے innerHTML کے اندر String کے فنکشن کو استعمال کرتے ہوئے ویریبیل x کو اس کے اندر لکھ دیا۔ اس سے اب x کی ویلو جو کہ نمبر ہے string کے فنکشن کے ساتھ اب string ویلو بن گئی ہے۔

```
<script>
    var x = 123;
    document.getElementById("numstr").innerHTML = String(x);
</script>
```

اس کا رزلٹ یہ ہے۔

123

اس مثال میں ہم نے وہی کام کیا لیکن اس مرتبہ ویرتبل بنانے کے بجائے نمبر ویلو کے ڈائریکٹ String کے فنکشن کے اندر لکھ دیا۔

```
<p id="numstr"></p>
```

```
<script>
```

```
document.getElementById("numstr").innerHTML = String(123);
```

```
</script>
```

اس کارزلٹ یہ ہے۔

123

اس مثال میں ہم نے پچھلی مرتبہ کی طرح ایک نمبر ویلو دی اور اس مرتبہ ہم نے اس String کے فنکشن کے اندر جمع بھی کر دیا۔

```
<p id="numstr"></p>
```

```
<script>
```

```
document.getElementById("numstr").innerHTML = String(100 + 23);
```

```
</script>
```

اس کارزلٹ یہ ہے۔

123

Date and Time

JavaScript کے اندر ہم تاریخ اور وقت بھی اپنے پیج کے اندر show کر سکتے ہیں۔ اور اس کا استعمال کر کے ان سے مختلف کام بھی لئے جاسکتے ہیں۔ JavaScript کے اندر کوئی بھی تاریخ یا وقت شروع کرنے سے پہلے ہمیں ایک بنیادی فنکشن کا استعمال کرنا پڑتا ہے۔ جس کے لئے ایک فنکشن **new Date ()** کا استعمال ہوتا ہے۔ اس فنکشن سے ہی باقی کے تمام فنکشن ڈیٹا اٹھاتے ہیں اور ہمیں مطلوبہ معلومات دے دیتے ہیں۔ نیچے درج اس کا Syntax ہے۔

```
new Date ( );
```

اس کو دیکھیں کہ یہ کس طرح کا output دیتا ہے۔ یاد رکھیں آپ جیسا دیکھ رہے ہیں آپ کو ویسا ہی لکھنا ہے۔ اگر الفاظ چھوٹا ہیں تو چھوٹا لکھیں اور اگر بڑا ہیں تو بڑا لکھیں۔ فنکشن میں ذرا بھی فرق آپ کے لئے مشکل پیدا کر سکتا ہے۔ اور Error کا سبب بن سکتا ہے۔ **new Date ()** اس کو بالکل اسی طرح لکھنا ہے اگر ذرا بھی فرق ہوگا تو یہ کام نہیں کرے گا اور Error کا سبب بنے گا۔

```
var date = new Date ( );
```

```
document.write (date);
```

ہم نے پہلے ایک ویریبل بنایا date کے نام سے پھر اس میں ہم نے **new Date ()** کا فنکشن استعمال کیا اور پھر ہم نے date کے ویریبل کو اپنے پیج میں پرنٹ کر دیا۔

اس کا رزلٹ یہ ہے۔

```
Sun Sep 20 2015 22:28:21 GMT+0500 (Pakistan Standard Time)
```

اس کوڈ سے آپ کو ساری چیزیں ایک ہی جگہ دیکھ جاتی ہیں۔ اگر ان میں سے کوئی ایک چیز مثلاً صرف وقت چاہیے تو آپ مزید فنکشنز کا استعمال کرتے ہیں۔ جن کے بارے میں ہم آگے تفصیل سے پڑھیں گے۔ اس کوڈ کو لکھنے کے چار طریقے ہیں۔ پہلا آپ دیکھ چکے

ہیں جس میں صرف `()` `new Date` کے فنکشن کا استعمال ہوا ہے۔ بقایا کی طرف بھی ایک نظر ڈالتے ہیں پھر آگے بڑھتے ہیں۔
نیچے درج اس کا Syntax ہے۔

`new Date (year, month, day, hours, minutes, seconds, milliseconds);`

اس فنکشن میں 7 پیرامیٹرز (سال، مہینہ، دن، گھنٹے، منٹس، سیکنڈز، ملی سیکنڈز) ہوتے ہیں۔ یہ امر کی فارمیٹ ہے اور ان کو اسی ترتیب سے لکھا جاتا ہے۔ JavaScript کے اندر سال 1 جنوری 1970 سے شروع ہوتا ہے۔ لیکن ہم ان پیرامیٹرز کا استعمال کر کے کوئی بھی تاریخ اور وقت حاصل کر سکتے ہیں۔ نیچے کوڈ دیکھیں۔ لیکن چونکہ JavaScript کے اندر مہینہ، دن، گھنٹے، منٹ، سیکنڈز اور ملی سیکنڈز سب 0 سے شروع ہوتے ہیں۔ اسی لئے اس کے بعد دن پھر گھنٹے،

`var date = new Date (1991,2,06,01,30,30,0);`

`document.write (date);`

اس کوڈ میں ہم نے `()` `new Date` کے اندر سال جو کہ ہم یہاں 1991 دے رہے ہیں۔ پھر اس کے بعد مہینہ جو کہ مارچ کا ہے اور اس کے لئے ہم نے 2 دیا ہے کیونکہ صفر سے شروع ہونے پر 2 نمبر مارچ کا ہوگا۔ اس کے بعد ہم نے تاریخ رکھی ہے جو کہ چھ تاریخ ہے۔ اس کے بعد ہم نے گھنٹے میں 1 اور منٹ میں 30 منٹ کا وقت رکھا ہے۔ سیکنڈز میں 30 سیکنڈز اور ملی سیکنڈز کو صفر رکھا ہے۔ ویسے سیکنڈز اور ملی سیکنڈز کو نہ بھی دیں تو کوئی فرق نہیں پڑے گا۔ لیکن آپ لوگوں کو سمجھانے کے لئے لکھ دیا ہے۔ اب ہوگا یہ کہ ہمارے `new Date()` کے فنکشن میں ہم نے جو پیرامیٹرز دیے ہیں یہ اس دن کا بالکل صحیح حساب لگا کر اس دن کی صحیح تاریخ ہمیں دے دے گا۔ مثلاً نیچے رزلٹ میں آپ دیکھ سکتے ہیں کہ دن بدھ آ رہا ہے لیکن ہم نے نہیں بتایا کہ دن کون سا ہے اور فارمیٹ بھی وہ نہیں جو ہم نے دیا تھا۔ ہونا تو یہ چاہئے تھا کہ پہلے سال آتا پھر مہینہ پھر دن اور پھر گھنٹے، منٹ اور سیکنڈز وغیرہ۔ لیکن اس کے برعکس یہاں ہمارے پاس پہلے ہفتے کا دن آ رہا ہے اس کے بعد مہینہ پھر تاریخ پھر سال اور پھر وقت دے رہا ہے اور ساتھ ہی ساتھ یہ ہمیں ہمارے کمپیوٹر کے وقت سے ہمیں یہ بتا رہا ہے کہ یہ پاکستان کا اسٹینڈرڈ ٹائم ہے۔
اس کا رزلٹ یہ ہوگا۔

Wed Mar 06 1991 01:30:30 GMT+0500 (Pakistan Standard Time)

اس اگلی مثال سے آپ کو مزید واضح ہو جائے گا۔ یہاں ہم صرف تاریخ کے متعلق کام کر رہے ہیں اور وقت کا اس سے کوئی لینا دینا نہیں ہے۔ یہاں بھی ہم نے وہی پیرامیٹر استعمال کئے ہیں۔ یعنی پہلے سال پھر مہینہ اور آخر میں تاریخ دی ہے۔ نیچے کوڈ دیکھیں۔

```
var date = new Date (1991,2,06);
```

```
document.write (date);
```

اس کا آؤٹ پٹ یہ ہوگا۔ اس میں کچھ خاص فرق نہیں پڑا لیکن صرف وقت رات کے 12:00 بجائے ہیں۔ اور چونہیں گھنٹے کے فارمیٹ کی وجہ سے یہ 00:00:00 نظر آ رہا ہے۔

Wed Mar 06 1991 00:00:00 GMT+0500 (Pakistan Standard Time)

یہ () newDate کو لکھنے کا ایک اور طریقہ کار ہے۔ اس میں ہم ملی سیکنڈز کا استعمال کرتے ہیں۔ یہ کافی مشکل ہے لیکن کبھی کبھی کام آتا ہے۔ ایک دن میں 86,400,00 ملی سیکنڈز ہوتے ہیں۔

نیچے درج اس کا Syntex ہے۔

```
new Date (milliseconds)
```

چونکہ JavaScript کا سال 1 جنوری 1970 سے شروع ہوتا ہے اسی لئے اگر آپ صرف 86,400,00 ملی سیکنڈز لکھتے ہیں تو آپ کو وہی تاریخ ملے گی بس آپ کے کمپیوٹر میں وقت UTC ہو۔ UTC اسٹینڈرڈ ٹائم کے مطابق اس وقت کیم جنوری 1970 ہو گا اور دن جمعرات کا ہو گا اور وقت دو بج کر چونہیں منٹ ہو رہے ہوں گے۔ لیکن اگر آپ کے کمپیوٹر میں اپنے علاقہ کا ٹائم سیو ہوا جیسے میرے کمپیوٹر میں پاکستان کا وقت سیو ہے تو اس صورت میں میری طرح آپ کے پاس بھی یہی رزلٹ آئے گا جو میرے پاس آ رہا ہے۔

```
var date = new Date (8640000);
```


`document.write (date);`

اس کا رزلٹ پاکستان اسٹینڈرڈ ٹائم کے مطابق یہ ہوگا۔

Fri Jan 02 1970 05:00:00 GMT+0500 (Pakistan Standard Time)

اور اس کا رزلٹ UTC اسٹینڈرڈ ٹائم کے مطابق یہ ہوگا۔

Thu Jan 01 1970 02:24:00 GMT+0000 (Coordinated Universal Time)

اب آجاتے ہیں ڈیٹ اسٹرنگ کی طرف اس کا استعمال گھڑی یا کیلنڈر بنانے میں ہوتا ہے۔ ڈیٹ اسٹرنگ دراصل ایک فنکشن ہے جو `new Date ()` کے فنکشن کے ساتھ کام کرتا ہے۔ نیچے درج اس کا Syntex ہے۔

`new Date (dateString)`

نیچے درج کوڈ کے اندر ہم نے `d` کے نام سے ایک ویریبل بنایا جس کو ہم نے `new Date()` کی مائنڈ دی اس کے بعد ہم نے ایک اور `d` کا ویریبل بنایا اسی نام سے کنفیوژ نہ ہوں JavaScript کے اندر ہم ایک نام کے مزید ویریبلز بھی بنا سکتے ہیں یہ عام بات ہے۔ اب ہم نے اپنے اگلے ویریبل کو پہلے والے `d` کے ویریبل کے ساتھ `toString()` کے فنکشن کے ساتھ کنکٹ کر دیا جس کے بعد ہمارے پہلے والے `d` کے ویریبل کی ساری چیزیں اب اگلے `d` کے ویریبل کے پاس آگئی۔ اب ہو گا یہ کہ پہلے `d` کا ویریبل `new Date()` کے فنکشن سے ساری تاریخ اٹھالے گا اور پھر اسے اگلے `d` کے ویریبل کے اندر `toString()` کے ساتھ مل کر اسے جو ویلیو درکار ہے دے دے گا اور بعد میں `document.write()` ہماری مظلوم `toString()` کی ویلیو جو `d` کا ویریبل ہمیں دے رہا ہے پرنٹ کر دے گا۔ اس کو مزید سمجھنے کے لئے یہ کوڈ دیکھیں۔

`d = new Date();`

`d = d.toString();`

```
document.write( d );
```

اس کارزلٹ ایسا ہوگا۔

Mon Sep 21 2015 06:54:45 GMT+0500 (Pakistan Standard Time)

اس اگلے کوڈ میں بھی ہم وہی کام کر رہے ہیں۔ سارا سب کچھ وہی ہے بس ہم `toString()` کے فنکشن کی جگہ `toDateString()` کے فنکشن کا استعمال کر رہے ہیں۔ جس سے اب ہمارے پاس صرف مہینہ، دن، تاریخ اور سال آئے گا۔ اس کو مزید سمجھنے کے لئے نیچے کوڈ دیکھیں۔

```
d = new Date( );  
d = d.toDateString( );  
document.write( d );
```

اس کارزلٹ ایسا ہوگا۔

Mon Sep 21 2015

یہ بھی ویسا ہی کوڈ ہے اس سے ہمیں UTC ٹائم ملتا ہے۔ اس کو مزید سمجھنے کے لئے نیچے کوڈ دیکھیں۔

```
d = new Date( );  
d = d.toUTCString( );  
document.write( d );
```

اس کارزلٹ ایسا ہوگا۔

Mon, 21 Sep 2015 01:52:37 GMT

اس کوڈ سے ہمیں ہمارے کمپیوٹر کا صرف وقت ملتا ہے اس کا زیادہ استعمال گھڑی بنانے میں ہوتا ہے اس کام کے لئے ہمیں صرف `toLocaleTimeString()` کے فنکشن کا استعمال کرنا ہوتا ہے۔ اس کو مزید سمجھنے کے لئے نیچے کوڈ دیکھیں۔

```
d = new Date();  
d = d.toLocaleTimeString();  
document.write( d );
```

اس کا رزلٹ ایسا ہوگا۔

6:56:55 AM

Date

اب یہاں تک تو بات ہو گئی؛ `new Date()` کے فنکشن کی بنیاد کی لیکن جیسا کہ آپ دیکھ سکتے ہیں اس میں ہمیں اب بھی تاریخ، دن، سال، مہینہ، منٹ، سیکنڈ اور ملی سیکنڈز سب ساتھ ہی مل رہے ہیں۔ لیکن اگر ہمیں سب کو الگ الگ کرنا ہو تو ہم کیا کریں گے یا یوں کہیں گے ہمیں اس میں سے صرف ایک چیز درکار ہے جیسے اگلی مثال میں ہمیں وقت درکار ہے تو ہم کیا کریں اب اس کام کے لئے ہم آچکے ہیں `get` اور `set` کی طرف سب سے پہلے `get` کی طرف چلتے ہیں۔

سب سے پہلے ہمیں تاریخ چاہئے اس کام کے لئے ہم `() getDate` کا ایک فنکشن استعمال کریں گے۔ یہ دو الفاظ `get` اور `Date` کا مجموعہ ہے۔ جن کے ملنے سے یہ فنکشن بنا ہے۔ یہ JavaScript کے اندر `new Date()` سے صرف مہینہ کے دن اٹھاتا ہے اور ہم اس کا استعمال صرف اپنے پیچ میں مہینہ کے دن شو کرنے کے لئے یا مہینہ کے دنوں سے متعلق کسی پروگرامنگ کے لئے کرتے ہیں۔ اس کے دو پیرامیٹر ہوتے ہیں جو کہ 1 سے لے کر 31 تک ہوتے ہیں جن سے ہم اپنی مرضی کی تاریخ بھی تیار کر سکتے ہیں۔

```
var date = new Date ();
var Date = date.getDate ();
document.write(Date);
```

یہاں پر آپ دیکھ سکتے ہیں کہ میں نے ایک ویریل بنایا `date` کے نام سے پھر اس کے اندر میں نے `{ } new Date` کا فنکشن استعمال کیا۔ اس سے اب یہ ہمیں پوری تاریخ ایک ساتھ مل گئی۔ پھر اس کے بعد ہم نے ایک اور ویریل بنایا `Date` کے نام سے اور اس کو ہم نے `date` کے ویریل کی ویلو دے دی۔ یعنی ہم نے پہلے والے ویریل کو دوبارہ دوسرے والے ویریل کے اندر دوبارہ استعمال کیا۔ جس سے اب ہم اس کے اندر سے کچھ بھی نکال سکتے ہیں۔ جیسا کہ آپ جانتے ہیں کہ جاوا اسکرپٹ ایک `Case Sensitive` زبان ہے اور اس میں ایک حرف کا فرق بھی اہمیت رکھتا ہے۔ اسی لئے یہ ایک مثال بھی ہے اور آپ دیکھ بھی سکتے ہیں کہ ہم نے ایک ہی نام کے دو ویریل بنائے صرف پہلے `date` چھوٹا تھا اور دوسرے `Date` بڑا اور اس سے بھی یہ دو الگ الگ ویریل بن گئے۔ اور اس کام کے لئے ہم `get` کا فنکشن استعمال کریں گے۔ لیکن جیسا کہ میں پہلے بھی بتا چکا ہوں کہ ہم ایک سے زیادہ فنکشنز کو ایڈ کرنے کے لئے `dot(.)` کے سائن کا استعمال کرتے ہیں۔ اسی لئے ہم یہاں بھی ایسا ہی کرنے والے ہیں۔ اس کام کے لئے ہم نے `date` کے بعد ایک `dot(.)` لگایا اور پھر `getDate ()` کا فنکشن استعمال کیا۔ یہاں ہم دنوں کے بارے میں معلوم کرنا چاہتے ہیں۔ اب اس کوڈ سے ہمارے پاس صرف اس دن کی تاریخ آئے گی جو ہمیں چاہیے۔

اس کا رزلٹ یہ ہوگا۔

اس مثال میں ہم سال کے مہینوں کے بارے میں معلوم کریں گے۔ اس کام کے لئے بھی ہم `{ } new Date` کے فنکشن کا استعمال کریں گے کیونکہ یہ بنیاد ہے اسی سے ہمیں ساری تاریخ ملتی ہے۔ اب مہینے کے بارے میں معلوم کرنے کے لئے ہم پھر اسی طریقے کا استعمال کریں گے جس کا استعمال ہم نے مہینے کے دن معلوم کرنے کے لئے کیا تھا۔ لیکن یہاں ہم مہینے کے لئے

getMonth() کا استعمال کریں گے۔ اس کے بھی دو پیرامیٹرز ہوتے ہیں جو کہ 0 سے لے کر 11 تک ہوتے ہیں جن سے ہم اپنی مرضی کی تاریخ بھی تیار کر سکتے ہیں۔ نیچے درج کوڈ دیکھیں پھر تفصیل میں آتے ہیں۔

```
var date = new Date ( ) ;  
var month = date.getMonth ( )+1 ;  
document.write(month) ;
```

یہاں بھی آپ دیکھ سکتے ہیں کہ ہم نے date کے ویریبل کے اندر newDate() کا فنکشن استعمال کیا پھر ایک اور ویریبل بنایا month کے نام سے اور اس میں ہم نے date کے ویریبل کے ساتھ getMonth() کے فنکشن کو کوئٹڈینٹ کر دیا۔ ابھی تو یہ صرف مہینہ کا نمبر ہی پرنٹ کرے گا۔ لیکن آپ switch Case if else کا استعمال کر کے انہیں String ویلو میں بھی پرنٹ کروا سکتے ہیں۔ اس کوڈ کے استعمال سے آپ مہینہ معلوم کر سکتے ہیں۔ لیکن اس میں ایک مشکل یہ ہے کہ JavaScript کے اندر مہینے، اور وقت کے گھنٹے، منٹ، سیکنڈ وغیرہ سب کے سب صفر 0 سے شروع ہوتے ہیں۔ جس کا مطلب ہے کہ اگر مہینہ ساتواں ہے تو آپ کا جواب 6 آئے گا۔ مثلاً دسمبر کا مہینہ بارہواں ہوتا ہے لیکن JavaScript کے اندر یہ گیارہواں نمبر پر آئے گا اور جنوری پہلا ہوتا ہے لیکن یہ صفر نمبر پر آئے گا۔ اس کا حل یہ ہے کہ آپ اس کے اندر ایک عدد جمع کر دیں۔ لیکن یاد رکھیں کہ جیسا لکھا ہے آپ کو بالکل ویسا ہی لکھنا کہ ورنہ کوڈ غلط ہوگا۔

اس کا رزلٹ یہ ہوگا۔

سال کے لئے بھی ہم اسی طریقہ کار کا استعمال کریں گے جس کا استعمال ہم نے دن اور مہینے کے لئے کیا تھا۔ اس کے چار پیرامیٹرز ہوتے ہیں جو کہ چار ہندسوں پر مشتمل ہوتے ہیں۔ جن سے ہم اپنی مرضی کی تاریخ بھی تیار کر سکتے ہیں۔ یہاں ایک بات اور ذہن نشین رکھ لیں کہ JavaScript کے اندر سال 1970 سے ہوتا ہے۔ اس سے پہلے کا سال آپ کو خود بنانا پڑے گا۔

```
var date = new Date ( );
var year = date.getFullYear ( );
document.write (year);
```

یہاں بھی آپ دیکھ سکتے ہیں کہ ہم نے date کے ویریبل کے اندر; newDate() فنکشن استعمال کیا پھر ایک اور ویریبل بنایا year کے نام سے اور اس میں ہم نے date کے ویریبل کے ساتھ; getFullYear() فنکشن کو کوئڈینٹ اینٹ کر دیا۔ یہ بھی دوسرے فنکشن کی طرح ہی کام کرے گا۔ اس کوڈ کے استعمال سے ہمیں سال مل جائے گا۔ لیکن یاد رہے کہ جیسا ہم نے کوڈ لکھا ہے ویسا ہی لکھنا ہے ورنہ غلط ہوگا۔ سال کے اس کوڈ کا استعمال آپ اپنی ویب سائٹ میں بھی کر سکتے ہیں جس سے آپ کی ویب سائٹ میں سال خود بخود اپ ڈیٹ ہو جائے گا۔

اس کا رزلٹ یہ ہوگا۔

2015

Day

اسی طرح ہفتے کے دنوں کے لئے بھی ہم ایک فنکشن; getDay() کا استعمال کریں گے۔ اس کے ساتھ پیرامیٹرز ہوتے ہیں جو کہ 0 سے لے کر 6 تک ہوتے ہیں جن سے ہم اپنی مرضی کے دن کی معلومات حاصل کر سکتے ہیں اور اس کا استعمال بھی کر سکتے ہیں۔ نیچے درج کوڈ دیکھیں پھر تفصیل میں آتے ہیں۔

```
var date = new Date ( );
var day = date.getDay();
document.write (day);
```

بر بار کی طرح یہاں بھی آپ دیکھ سکتے ہیں کہ ہم نے date کے ویریبیل کے اندر: () newDate() فنکشن استعمال کیا پھر ایک اور ویریبیل بنایا day کے نام سے اور اس میں ہم نے date کے ویریبیل کے ساتھ: () getDay() کے فنکشن کو کنکلیڈینٹ کر دیا۔ اس سے اب ہمیں ہفتے کا جو دن ہوگا وہ مل جائے گا۔ اس میں بھی ہفتے کا دن 0 سے شروع ہوتا ہے۔ جن کے بارے میں آپ switch Case میں پہلے ہی دیکھ چکے ہیں۔ یعنی اگر نمبر 6 ہوگا تو دن ہفتے کا ہوگا۔

اس کا رزلٹ یہ ہوگا۔

6

Hour

اب آتے ہیں گھنٹے کی طرف ایک دن میں چوبیس گھنٹے ہوتے ہیں۔ اور گھنٹوں کو ویب پیج پر شو کرنے کے لئے بھی ہم ایک فنکشن: () getHours() کا استعمال کرتے ہیں۔ اس کے بھی دو پیرامیٹرز ہوتے ہیں جو کہ 0 سے لے کر 23 تک ہوتے ہیں۔ یہ چوبیس گھنٹے کے فارمیٹ پر ہوتا ہے۔ لیکن ہم if اور else کا استعمال کر کے اسے AM اور PM میں بھی کنورٹ کر سکتے ہیں۔ نیچے درج کوڈ دیکھیں۔

```
var time = new Date ( );
```

```
var hour = time.getHours( );
```

```
document.write(hour);
```

اس کوڈ میں ہم نے سب سے پہلے ایک time کے نام سے ویریبیل بنایا اور پھر اسے: () newDate() کی مائڈ وی اس کے بعد ایک اور ویریبیل بنایا hour کے نام سے اور اس کو time کے ویریبیل کے ساتھ کنکلیڈینٹ کر کے: () getHours() کے فنکشن کا استعمال کیا۔ اور پھر: () document.write() کے ذریعے اسے پرنٹ کر دیا۔

اس کا رزلٹ دیکھیں۔

7

میرے کمپیوٹر میں اس وقت صبح کے ساتھ جا رہے ہیں اسی لئے یہ نمبر سات دے رہا ہے لیکن اگر میرے کمپیوٹر میں دوپہر کے ایک بجے ہوتے تو اس وقت یہ 13 دیتا کیونکہ جیسا میں پہلے ہی بتا چکا ہوں کہ یہ چوبیس گھنٹے کے فارمیٹ پر ہوتا ہے۔ اب اسے 12 گھنٹوں کے فارمیٹ پر لانے کے لئے ہم if اور else کا استعمال کریں گے۔ لیکن پہلے ہم AM اور PM لکھیں گے ورنہ بعد میں ہمارا کوڈ غلط ہو جائے گا۔

```
var time = new Date ( );
var hour = time.getHours( );
document.write( hour + " " +sort );

if (hour < 12){
    sort = "AM";
}

else (sort="PM");
```

یہاں ہم نے ایک if کنڈیشن استعمال کی اور اس میں کنڈیشن دی کہ اگر hour کا ویریبل کم ہو 12 سے تو پھر ہم نے ایک sort کے نام سے ایک ویریبل بنایا اور اس میں ایک اسٹرنگ ویلو دے دی AM کی جس سے اب جیسے ہی ہمارا وقت 12 گھنٹوں سے کم ہو گا وہ AM ہو جائے گا۔ اور پھر ہم نے else کا استعمال کیا اور اس میں sort کی ویلو PM کر دی۔ جس سے اب جیسے ہی وقت 12 گھنٹوں سے اوپر جائے گا۔ اس کی ویلو PM ہو جائے گی۔ اور اس کو میں نے document.write کے اندر hour کے ویریبل کے ساتھ پرنٹ کر دیا۔ آپ ان دونوں کے درمیان میں ڈبل کوٹیشن اور ان کے درمیان ایک اسپیس دیکھ رہے ہوں گے اس سے ان دونوں کے درمیان جھوڑا فاصلہ آجائے گا۔

اس کا زرلٹ کچھ اس طرح ہو گا۔

یہ غلط ہے 13 بجے PM نہیں آتا اب اس کو 12 گھنٹوں کے فارمیٹ میں تبدیل کرنے کے لئے ہم if کا استعمال کریں گے۔

```
var time = new Date ( );
var hour = time.getHours( );
document.write( hour + " " +sort );
if (hour < 12){
    sort = "AM";
}
else (sort="PM");
if(hour > 12){
    hour=hour-12;
}
```

ہم نے یہاں ایک if کنڈیشن بنایا اور اس میں کنڈیشن لکھی کہ اگر hour کا ویرتبل 12 سے زیادہ ہو تو hour کے ویرتبل کے اندر hour کے ویرتبل کو ہی اسائن کر دو اور اس میں سے 12 کو مائنس کر دو اس طرح سے اب اوپر والے ویرتبل کا تعلق ختم اب hour کے ویرتبل کے پاس جو ویلو جائے گی وہ ہمارے if کے کنڈیشن کے hour کے ویرتبل کی ویلو جائے گی اب ہمارے پاس جو ویلو آئے گی وہ کچھ اس طرح ہوگی۔

1 PM

Minutes

اسی طرح ایک گھنٹے میں 60 منٹ ہوتے ہیں جن کو شو کرنے کے لئے بھی ہم (); getMinutes کے کنڈیشن کا استعمال کریں گے۔ اس کے بھی دو پیرامیٹرز ہوتے ہیں جو 0 سے لے کر 59 تک ہوتے ہیں۔

```
var time = new Date ();
```

```
var minute = time.getMinutes();
```

```
document.write (minute);
```

یہاں آپ دیکھ سکتے ہیں کہ ہم نے پہلے time کے نام سے ایک ویریبل بنایا اور اس کو یلڈی؛ newDate() کی پھر ایک اور ویریبل بنایا minute کے نام سے اور اس کو time کے ویریبل کے ساتھ؛ getMinutes() کے فنکشن کے ساتھ کنکٹڈ یٹڈ کر دیا پھر اس کے بعد ہم نے document.write کے اندر minute کے ویریبل کو پرنٹ کر دیا۔

اس کا رزلٹ یہ ہوگا۔

5

یہاں آپ دیکھ سکتے ہیں کہ 5 منٹ ہوئے ہیں اور صرف 5 نظر آ رہا ہے جو کہ گھڑی میں بالکل بھی اچھا نہیں لگے گا۔ ہمیں 5 کے آگے ایک 0 کی ضروری ہے۔ اس کام کے لئے ہم ایک if کے فنکشن کا استعمال کریں گے۔

```
if (minute < 10){
```

```
minute="0"+minute;
```

```
}
```

آپ دیکھ سکتے ہیں کہ ہم نے پہلے if کا ایک فنکشن بنایا جس کو کنڈیشن دی کہ اگر minute کا ویریبل کم ہو 10 سے تو اس minute کے ویریبل کو 0 کر دیں اور آپ دیکھ سکتے ہیں کہ 0 ایک اسٹرنگ ویلو بنایا ہوا ہے کیوں کہ ہمیں اسے ایسا ہی پرنٹ کروانا ہے۔ اسے اسٹرنگ ویلو بنانے کے لئے ہم نے 0 کو ڈبل کوٹیشن کے اندر لکھا ہے۔ پھر ہم نے منٹ کے ویریبل جس کو ہم نے 0 کیا تھا اس کے اندر پھر سے ہم نے minute کے ویریبل کا اضافہ کر دیا۔ اب ہو گا یہ کہ جب ہمارا minute کا ویریبل پرنٹ ہو گا تو پہلے جسے ہم نے 0 کیا تھا وہ منٹس کو 0 کر دے گا پھر جو ہم نے + کے سائن کے ساتھ دوبارہ minute کے ویریبل کا اضافہ کیا تھا اس سے ہمارے پرنٹ یا گھڑی میں منٹس دوبارہ شو ہو جائیں گے۔ لیکن اس بار اس کے آگے ایک 0 بھی ہوگا۔ ہم اس کام کے لئے 0 کو ڈبل کوٹ بھی لکھ

سکتے تھے لیکن پھر اس صورت میں جب منٹس 10 یا اس سے زیادہ ہوتے اور بھی 0 ان کے آگے ہی شو ہوتا جو بالکل اچھا نہیں لگتا اسی لئے ہمیں یہ کوڈ لکھنا پڑا۔
اس کارزلٹ یہ ہوگا۔

06

Seconds

منٹس کی طرح ایک سیکنڈ میں بھی 60 ملی سیکنڈز ہوتے ہیں اور ان کے لئے ہم (getSeconds()) کے فنکشن کا استعمال کرتے ہیں۔ اس کے بھی دو ہی پیرامیٹرز ہوتے ہیں جو کہ 0 سے لے کر 59 تک ہوتے ہیں۔

```
var time = new Date ();
var second = time.getSeconds();
document.write (second);
```

یہاں بھی آپ دیکھ سکتے ہیں پہلے والا ہی کوڈ ہے بس minute کے ویریبیل کی جگہ second کا ویریبیل ہے اور getMinutes() کی جگہ getSeconds() ہے۔

اس کارزلٹ یہ ہوگا۔

1

اور یہاں بھی وہی صورت حال ہے سیکنڈ بھی اکیلا ہی آئے گا۔ اس کے آگے ہی ہمیں 0 کے لئے وہی if کا فنکشن استعمال کرنا ہوگا بس اس بار minute کی جگہ second استعمال کرنا ہوگا۔

```

if (second < 10){
second="0"+second;
}

```

اس کا رزلٹ یہ ہوگا۔

09

Milli Seconds

ملی سیکنڈز کے پیرامیٹرز تو تین ہوتے ہیں لیکن یہ ہوتا چار ہندسوں والا ہے۔ ایک سیکنڈ میں ایک ہزار ملی میٹر ہوتے ہیں۔ اس کے لئے ہم `getMilliseconds()` کے فنکشن کا استعمال کریں گے۔ اس کے پیرامیٹرز 0 سے لے کر 999 تک ہوتے ہیں۔

```

var date = new Date ();
var milisecond = date.getMilliseconds();
document.write (milisecond);

```

اس کا رزلٹ ایسا ہوگا۔ کیونکہ اب جتنی دیر میں ویب پیج کو ریفریش کریں گے اتنی دیر میں دو سو یا ڈھائی سو ملی سیکنڈز گزر چکے ہوں گے۔

563

آپ کی آسانی کے لئے کچھ بنیادی فنکشنز کا ایک چارٹ موجود ہے جس سے آپ کو پریشانی نہیں ہوگی۔

تفصیل	فکشن
مینے کے دن جو کہ 1 سے لے کر 31 تک ہونگے۔ (1-31)	newDate()
ہفتے کے دن جو کہ 0 سے شروع ہو کر 6 تک ہونگے۔ (0-6)	getDay()
مینے جو کہ 0 سے 11 تک ہونے۔ (0-11)	getMonth()
سال جو کہ چار ہندسوں میں ہوگا۔ (yyyy) (1970)	getFullYear()
گھنٹے جو کہ 0 سے 23 تک ہونگے۔ (0-23)	getHours()
سیکنڈ جو کہ 0 سے 59 تک ہونگے۔ (0-59)	getSeconds()
منٹ جو کہ 0 سے لے کر 59 تک ہونگے۔ (0-59)	getMinutes()
میلی سیکنڈ جو کہ 0 سے لے کر 999 تک ہونگے۔ (0-999)	getMilliseconds()
وقت جو کہ JavaScript کے شروع وقت یعنی 1 جنوری 1970 سے ہوگا۔ جس میں امریکی اسینڈرڈ کے مطابق مہینہ پہلا پھر دن اور آخر میں سال آئے گا۔ (January 1, 1970)	getTime()

Cookie

یہاں ہم Cookie کے بارے میں پڑھیں گے۔ یہ کھانے والی Cookie نہیں بلکہ اس کا استعمال ہم یا داشت کے طور پر کرتے ہیں۔ یہ بھی دراصل BOM کا حصہ ہے۔ cookie کے دو بنیادی کام ہوتے ہیں۔ پہلا یوزر کی انفارمیشن حاصل کرنا اور دوسرا جب یوزر دوسری بار پیج وزٹ کرے تو اسے یاد رکھنا۔ مثال کے طور پر جب آپ فیس بک یا اپنی کسی ای میل پر لوگ ان کرتے ہیں تو remember me کا ایک بکس ہوتا ہے اگر آپ اسے سلیکٹ کر لیتے ہیں تو وہ آپ کے کمپیوٹر میں ایک فائل بنا دیتی ہے اور اگر آپ لوگ آؤٹ نہ کریں تو جب آپ دوبارہ وہ فائل وزٹ کرتے ہیں تو آپ کو دوسری بار لوگ ان کرنے کی ضرورت نہیں رہتی وہ دوبارہ ایسے ہی اوپن ہو جاتی ہے۔ cookie ایک مخصوص وقت کے لئے ہوتی ہے اور پھر اس کے بعد خود بہ خود ڈیلیٹ ہو جاتی ہے۔

cookie کو بنانے کے لئے ہم document.cookie کا استعمال کرتے ہیں۔

مثال:

```
document.cookie="username=Danish Irshad; expires=Thu, 18 Dec 2015  
12:00:00 UTC";
```

cookie کو پڑھنے کے لئے بھی ہم document.cookie کا استعمال کرتے ہیں۔

مثال:

```
var x = document.cookie;  
document.write( d );
```

cookie کو تبدیل کرنے کے لئے بھی ہم document.cookie کا استعمال کرتے ہیں۔ اس سے تبدیل شدہ cookie پرانی cookie سے تبدیل ہو جاتی ہے۔

مثال:

```
document.cookie="username=document.cookie=username=Muhammad  
Danish Irshad; expires=Thu, 18 Dec 2015 12:00:00 UTC";
```

cookie کو ختم کرنے کے لئے بھی ہمیں صرف اس کی expire date کو تبدیل کرنا ہے۔

مثال:

```
document.cookie = "username=; expires=Thu, 01 Jan 1970
00:00:00 UTC";
```

DOM

Document Object Model یا DOM دراصل ویب میں موجود ڈاکیومنٹس کے لئے ہوتے ہیں۔ ویب پیج جیسے ہی لوڈ ہوتا ہے یہ اپنا کام شروع کر دیتا ہے۔ عام طور پر اس کو کسی لفظ کو تبدیل کرنے یا پرنٹ کروانے کے لئے استعمال کیا جاتا ہے۔ ہم اس کو دو حصوں میں تقسیم کر سکتے ہیں۔ مثال کے لئے آپ نیچے والا کوڈ دیکھیں۔

```
<p id="demo">Hello</p>
```

```
<script>
```

```
document.getElementById("demo").innerHTML = "Hello World";
```

```
</script>
```

اس کوڈ کے اندر ہم نے پہلے ایک پیراگراف کا ٹیگ استعمال کیا جس کو ہم نے id دی demo کے نام سے پھر ہم نے Script کے ٹیگ کے اندر اپنا کوڈ لکھا۔ پہلے ہم نے document.getElementById() کا کوڈ استعمال کیا جس سے ہمارے براؤزر کو پتا چلتا ہے کہ اب ہمیں ایک پتا کرنا ہے کہ کس ٹیگ کی id کا نام demo ہے۔ پھر ہم نے innerHTML کا کوڈ استعمال کیا جس کو ہم نے ویلو دے دی "Hello World" کی اسے ایک ویریبل کی طرح ہی ویلو دی جاتی ہے۔ کوڈ دیکھیں۔

```
innerHTML = "Hello World";
```

اور جیسے کہ پہلے بھی پڑھ چکے ہیں ان دونوں کوڈز کو ملانے کے لئے ہمیں dot (.) کا استعمال کرنا پڑتا ہے تو ہم نے ان دونوں کوڈز کے درمیان ایک dot (.) لگا دیا۔

اب پیراگراف کے اندر Hello کوڈ کو یہ کوڈ Hello World بنادے گا جیسے ہی ویب پیج لوڈ ہوتا ہے۔

اب جو زیادہ تر لوگوں کے ذہن میں سوال آتا ہے کہ یہ کام تو ہم ڈائریکٹ بھی کر سکتے تھے تو اتنی محنت کیوں؟

تو جناب جواب یہ ہے کہ اس کا استعمال ہم مختلف پوزیشن پر کرتے ہیں۔

DOM کے مزید Elements یہ ہیں ان سے آپ کو اس کو سمجھنے میں آسانی ہوگی۔

یہ Finding Elements کہلاتے ہیں۔

document.getElementById()

اس کوڈ سے ہم HTML ٹیگ کی id کو معلوم کر کے اس پر کام کرتے ہیں۔ اس کے بارے میں ہم پہلے ہی پڑھ چکے ہیں۔

document.getElementsByTagName()

اس کوڈ سے ہم HTML ٹیگ کا نام لکھتے ہیں اور پھر اس پر کام کرتے ہیں۔ لیکن اگر آپ کے پیج میں ایک ہی نام کے بہت سارے ٹیگ ہیں تو یہ سب پر کام کرے گا اسی لئے اس کو بہت کم استعمال کیا جاتا ہے۔

document.getElementsByClassName()

اس کوڈ سے ہم HTML ٹیگ کی Class کو معلوم کر کے اس پر کام کرتے ہیں۔ یہ id کی ہی طرح کام کرتا ہے۔

یہ Changing Elements کہلاتے ہیں۔

.innerHTML=

اس کا استعمال بھی ہم پہلے ہی کر چکے ہیں۔ یہ HTML کے اندر کسی بھی ٹیگ کے مواد کو تبدیل کرنے کے لئے استعمال ہوتا ہے۔

.attribute=

اس کا استعمال بھی ہم پہلے ہی کر چکے ہیں۔ یہ HTML کے اندر کسی بھی ٹیگ کے مواد کو تبدیل کرنے کے لئے استعمال ہوتا ہے۔

.setAttribute(attribute,value)

اس کا استعمال بھی ہم پہلے ہی کر چکے ہیں۔ یہ HTML کے اندر کسی بھی ٹیگ کے مواد کو تبدیل کرنے کے لئے استعمال ہوتا ہے۔

.style.property=

اس کا استعمال ہم JavaScript کے اندر CSS کی پروپرتی کیا استعمال کے لئے کرتے ہیں۔ اس کی پروپرتی مختلف ہو سکتی ہیں جن کو ہم = کے سائن کے بعد لکھتے ہیں۔

مثال کے طور پر =.style.font, .style.color وغیرہ

یہ Adding and Deleting Elements کہلاتے ہیں۔

document.createElement()

اس کا استعمال بھی ہم پہلے ہی کر چکے ہیں۔ یہ HTML کے اندر کسی بھی ٹیگ کے مواد کو تبدیل کرنے کے لئے استعمال ہوتا ہے۔

document.removeChild()

اس کا استعمال بھی ہم پہلے ہی کر چکے ہیں۔ یہ HTML کے اندر کسی بھی ٹیگ کے مواد کو تبدیل کرنے کے لئے استعمال ہوتا ہے۔

document.appendChild()

اس کا استعمال بھی ہم پہلے ہی کر چکے ہیں۔ یہ HTML کے اندر کسی بھی ٹیگ کے مواد کو تبدیل کرنے کے لئے استعمال ہوتا ہے۔

document.replaceChild()

اس کا استعمال بھی ہم پہلے ہی کر چکے ہیں۔ یہ HTML کے اندر کسی بھی ٹیگ کے مواد کو تبدیل کرنے کے لئے استعمال ہوتا ہے۔

document.write()

اس کا استعمال بھی ہم پہلے ہی کر چکے ہیں۔ یہ JavaScript کے اندر کسی بھی مواد کو ویب پیج پر پرنٹ کرنے کے لئے استعمال ہوتا ہے۔

BOM

Browser Object Model یا BOM دراصل براؤزر یا ڈیٹیکٹاپ کی انفارمیشن کے لئے ہوتا ہے۔ لیکن اس سے مختلف کام بھی لئے جاسکتے ہیں۔ ہم BOM کو سات حصوں میں تقسیم کر سکتے ہیں۔ جن میں سے دو حصوں Popup Boxes اور Cookies کے بارے میں ہم پہلے ہی پڑھ چکے ہیں اب مزید پانچ کے بارے میں پڑھیں گے۔

Window

ونڈو اسکرین کا استعمال ہم JavaScript کے اندر ونڈو کے لئے کرتے ہیں اس کے چند کوڈز درج ذیل ہیں۔ ویسے تو popup ونڈو بھی ونڈو ہی ہیں لیکن وہ window prefix کے بغیر لکھی جاتی ہیں اسی لئے انہیں الگ کیلگری میں رکھا جاتا ہے۔

window.open()

اس سے ہم نئی ونڈو اوپن کرنے کے کام آتا ہے۔

window.close()

اس سے ہم اوپن ونڈو کو بند کرنے کے کام آتا ہے۔

window.moveTo()

اس سے ہم اوپن ونڈو کو موو (move) کرنے کے کام آتا ہے۔

window.resizeTo()

اس سے ہم اوپن ونڈو کی سائز یعنی چھوٹا بڑا کرنے کے کام آتا ہے۔

window.innerHeight

اس سے ہم نی اوپن ونڈو کی اونچائی معلوم کر سکتے ہیں۔

window.innerWidth

اس سے ہم نی اوپن ونڈو کی لمبائی معلوم کر سکتے ہیں۔

document.body.clientHeight

اس سے ہم ویب پیج کی اونچائی معلوم کر سکتے ہیں۔

document.body.clientWidth

اس سے ہم ویب پیج کی لمبائی معلوم کر سکتے ہیں۔

Window Screen

اس کا استعمال ہم اسکرین کی معلومات اور سائزنگ کے لئے کرتے ہیں۔

screen.width

اس سے ہم اسکرین کی لمبائی معلوم کر سکتے ہیں۔

مثال:

```
<p id="demo"><p>
```

```
<script>
```

```
document.getElementById("demo").innerHTML = "Screen Width: " +
```

```
screen.width;
```

```
</script>
```

رزلٹ

Screen Width: 1280

screen.height

اس سے ہم اسکرین کی اونچائی معلوم کر سکتے ہیں۔

مثال:


```
<p id="demo"><p>
```

```
<script>
```

```
document.getElementById("demo").innerHTML = "Screen Height: " +
```

```
screen.height;
```

```
</script>
```

رزلٹ

Screen Height: 1024

screen.availWidth

اس سے ہم یہ اسکرین کی موجودہ لمبائی معلوم کر سکتے ہیں۔

مثال:

```
<p id="demo"><p>
```

```
<script>
```

```
document.getElementById("demo").innerHTML = "Available Screen Width: " +
```

```
screen.availWidth;
```

```
</script>
```

رزلٹ

Available Screen Width: 1280

screen.availHeight

اس سے ہم اسکرین کی موجودہ اونچائی معلوم کر سکتے ہیں۔

مثال:

```
<p id="demo"><p>
```

```
<script>
```

```
document.getElementById("demo").innerHTML = "Available Screen Height: "
```

```
+ screen.availHeight;
```

```
</script>
```

رزلٹ

Available Screen Height: 995

screen.colorDepth

اس سے ہمیں پتا لگتا ہے کہ ہماری اسکرین پر کتنے کلر گتے ہیں ایک کلر کو شو کرنے کے لئے مثلاً 24 بٹ میں 16,777,216 مختلف True کلر ہوتے ہیں۔ جبکہ 32 بٹ میں 4,294,967,296 مختلف Deep کلر ہوتے ہیں۔

مثال:

```
<p id="demo"><p>
```

```
<script>
```

```
document.getElementById("demo").innerHTML = "Screen Color Depth: " +
```

```
screen.colorDepth;
```

</script>

رزلٹ

Screen Color Depth: 24

screen.pixelDepth

اس پروپٹی کا استعمال ہم اسکرین کے پیکسل ڈپتھ کے معلوم کرنے کے لئے کرتے ہیں۔

مثال:

<p id="demo"><p>

<script>

document.getElementById("demo").innerHTML = "Screen Pixel Depth: " +

screen.pixelDepth;

</script>

رزلٹ

Screen Pixel Depth: 24

Window Location

ونڈو لوکیشن کا استعمال ہم موجودہ پیج کا URL یا اس پیج کو دوسری جگہ ری ڈائریکٹ کرنے کے لئے کرتے ہیں۔

window.location.href

اس کا استعمال ہم پیج کا URL معلوم کرنے کے لئے کرتے ہیں۔

مثال:

```
<p id="demo"><p>
<script>
document.getElementById("demo").innerHTML =
"Page location is " + window.location.href;
</script>
```

رزلٹ

Page location is
/F:/Danish%20Files/19-August-2015js/js_window_location.htm

window.location.hostname

اس کا استعمال ہم پیج کا Hostname معلوم کرنے کے لئے کرتے ہیں۔

مثال:

```
<p id="demo"><p>
<script>
```



```
document.getElementById("demo").innerHTML =
"Page hostname is " + window.location.hostname;

</script>
```

رزلٹ

Page hostname is

window.location.pathname

اس کا استعمال ہم پیج کا پاتھ معلوم کرنے کے لئے کرتے ہیں۔

مثال:

```
<p id="demo"><p>

<script>

document.getElementById("demo").innerHTML =

"Page path is " + window.location.pathname;

</script>
```

رزلٹ

/19-August-2015/js_window_location.htm

window.location.protocol

اس کا استعمال ہم پیج کا پرنٹ کوئل معلوم کرنے کے لئے کرتے ہیں۔

مثال:

```
<p id="demo"><p>
<script>
document.getElementById("demo").innerHTML =
"Page protocol is " + window.location.protocol;
</script>
```

رزلٹ

Page protocol is file:ftp://

Window History

ونڈو ہسٹری کا استعمال ہم ونڈو کی ہسٹری کے لئے کرتے ہیں۔

history.back()

اس کا استعمال ہم ونڈو کی ہسٹری سے اپنے پیج پر پیچھے جانے کے لئے کرتے ہیں۔

مثال:

```

<script>
function goBack() {
    window.history.back()
}
</script>

<input type="button" value="Back" onclick="goBack()">

```

رزلٹ



history.forward()

اس کا استعمال ہم ونڈ کی ہسٹری سے اپنے پیج پر آگے جانے کے لئے کرتے ہیں جس پیج کو پہلے ہی وزٹ کر چکے ہوں اور پھر پیچھے جا چکے ہوں۔

مثال:

```

<script>
function goForward() {
    window.history.forward()
}
</script>

<input type="button" value="Forward" onclick="goForward()">

```

رزلٹ

Forward

Window Navigator

ونڈو Navigator کے استعمال سے ہم براؤزر کی انفارمیشن حاصل کرتے ہیں۔

Navigator Cookie Enabled

اس کوڈ کے استعمال سے ہمیں پتا چلتا ہے کہ یوزر کے براؤزر میں cookie enable ہے یا نہیں۔
بہت ساری ویب سائٹس کوکی کا استعمال کرتی ہیں۔ اس کوڈ کے استعمال سے اگر براؤزر میں cookie enable نہ ہو تو یہ کوڈ بتا دیتا ہے اور اس صورت میں یوزر کو آپ اپنی ویب سائٹ پر آنے سے روک سکتے ہیں یہ اس کے لئے مخصوص چیزیں رکھ سکتے ہیں۔

مثال:

```
<p id="demo"><p>
```

```
<script>
```

```
document.getElementById("demo").innerHTML =
```

```
"Cookies Enabled is " + navigator.cookieEnabled;
```

```
</script>
```

رزلٹ

Cookies enabled is true

Browser Names

اس کا استعمال کر کے ہم براؤز کا نام معلوم کر سکتے ہیں کہ یوزر کون سا براؤزر استعمال کر رہا ہے۔

نوٹ: Internet Explorer 11, Chrome, Firefox, اور Safari کا appName Netscape ہوگا۔

appCodeName Mozilla کا Opera اور Chrome, Firefox, Internet Explorer, Safari,

ہوگا۔

مثال:

```
<p id="demo"><p>
```

```
<script>
```

```
document.getElementById("demo").innerHTML = "Name is " +  
navigator.appName + ". Code name is " + navigator.appCodeName;
```

```
</script>
```

رزلٹ

Name is Netscape
Code name is Mozilla

Browser Engine

اس سے ہم براؤزر کے انجن کے بارے میں معلومات حاصل کرتے ہیں۔

مثال:

```
<p id="demo"><p>
```

```
<script>
```

```
document.getElementById("demo").innerHTML = navigator.product;
```

```
</script>
```

رزلٹ

Browser engine is Gecko

Browser appVersion

اس سے ہم براؤزر کے appVersion کے بارے میں معلومات حاصل کرتے ہیں۔

مثال:

```
<p id="demo"><p>
```

```
<script>
```

```
document.getElementById("demo").innerHTML = navigator.appVersion;
```

```
</script>
```

رزلٹ

**5.0 (Windows NT 5.1) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/46.0.2490.71 Safari/537.36**

Browser userAgent

اس سے ہم براؤزر کے userAgent کے بارے میں معلومات حاصل کرتے ہیں۔

مثال:

```

<p id="demo"><p>
  <script>
document.getElementById("demo").innerHTML = navigator.userAgent;
  </script>

```

رزلٹ

Mozilla/5.0 (Windows NT 5.1) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/46.0.2490.71 Safari/537.36

Browser Platform

اس سے ہمیں پتا چلتا ہے کہ براؤزر کس اوپریٹنگ سسٹم پر چل رہا ہے۔

مثال:

```

<p id="demo"><p>
  <script>
document.getElementById("demo").innerHTML = navigator.platform;

```

</script>

رزلٹ

Win32

Browser Language

اس سے ہمیں پتا چلتا ہے کہ ہمارے براؤزر کی زبان اس وقت کون سی استعمال ہو رہی ہے۔

مثال:

<p id="demo"><p>

<script>

document.getElementById("demo").innerHTML = navigator.language;

</script>

رزلٹ

Browser language is en-US

Java Enabled

اس سے ہمیں پتا چلتا ہے کہ ہمارے براؤزر میں Java Enable ہے یا نہیں۔

مثال:

```
<p id="demo"><p>
<script>
document.getElementById("demo").innerHTML = navigator.javaEnabled();
</script>
```

رزلٹ

Java enabled is true

Console.log();

اس کے استعمال سے ہم براؤزر کے console میں کچھ بھی لکھ سکتے ہیں۔
اسے دیکھنے کے لئے آپ chrome، firefox، internet explorer میں F12 کا بٹن دبائیں اور پھر console کا
اوپنیشن سلیکٹ کریں آپ کو لکھا دکھ جائے گا۔

```
console.log("hello earth");
```

اس کا رزلٹ یہ ہوگا۔



Timing Events

ٹائمنگ ایونٹس کا استعمال ہم مختلف کاموں میں کرتے ہیں۔ یہاں ہم setTimeout اور clearTimeout کے بارے میں پڑھیں گے۔

☆ setTimeout() اس کے استعمال سے ہم کسی بھی فنکشن کو ایک مخصوص وقت پر چلا سکتے ہیں اس کا وقت ہمیں ملی سیکنڈز میں دینا ہوتا ہے۔ جیسے 1000 میں ایک سیکنڈ

☆ setInterval() یہ بھی setTimeout کی طرح ہے لیکن یہ فنکشن ہولگا تا رہتا رہتا ہے۔

setTimeout() کے دو پیرامیٹرز ہوتے ہیں ایک میں ہم فنکشن کا نام لکھتے ہیں اور دوسرے میں ملی سیکنڈز لکھتے ہیں۔ اس کو مزید سمجھنے کے لئے کوڈ دیکھیں۔

```
<button onclick="setTimeout(myFunction, 3000);">
```

```
click and wait 3 Seconds</button>
```

یہاں ہم نے ایک بٹن بنایا اور اس کے اندر onclick کے Attribute کا استعمال کیا اور اس کو ہم نے setTimeout() کی ویلو دی اور setTimeout کے اندر ہم نے Function کا نام جو کہ ہمارے پاس myFunction کے نام سے ہے لکھا اور دوسرے پیرامیٹر میں 3000 ملی سیکنڈز لکھ دیا۔ اور detail میں click and wait 3 Seconds لکھ دیا۔

```
<script>
```

```
function myFunction() {
```

```
document.write('Hello World');
```

```
}
```

```
</script>
```

اس لائن میں ہم نے Script کے ٹیگ کے اندر ایک فنکشن بنایا myFunction کے نام سے اور اس کے اندر ایک document.write کا فنکشن استعمال کیا اور اس کے اندر ہم نے ایک اسٹرنگ ویلڈو Hello World پرنٹ کروایا۔ اب اس سے ہوگا یہ کہ جب کوئی یوزر بٹن پر کلک کرے گا اور 3 سیکنڈز انتظار کرے گا تو اسے 3 سیکنڈز کے بعد ہیچ پر Hello World کا پیغام ملے گا۔

اس کا رزلٹ یہ ہوگا۔

کلک کے بعد	کلک سے پہلے
Hello World	click and wait 3 Seconds

اس مثال میں ہم نے دو بٹن بنائیں ہیں ایک سے ہمارا فنکشن چلے گا اور دوسرے سے ہمارا فنکشن بند ہو جائے گا۔ فنکشن بند کرنے کے لئے ہم clearTimeout کا استعمال کریں گے۔

```
<button onclick="myVar = setTimeout(myFunction, 3000)">Start</button>
```

```
<button onclick="clearTimeout(myVar)">Stop</button>
```

یہاں ہم نے دو بٹن بنائیں ہیں پہلے والے بٹن میں onclick کے event کے اندر ایک ویریبیل بنایا myVar کے نام سے اور اس کو ویلڈو دے دیں۔ setTimeout کی اور اس کے اندر فنکشن کا نام myFunction اور 3000 ملی سیکنڈز دے دیے اور details کے لئے start لکھ دیا جس سے ہمیں پتا چلے گا کہ اس سے ہمارا فنکشن start ہوگا۔ دوسرے والے بٹن میں ہم نے onclick کے Attribute کے اندر clearTimeout کا استعمال کیا ہے اور اس کے اندر ہم نے جو ویریبیل پہلے والے بٹن میں بنایا تھا myVar کے نام سے اسی ویریبیل کا نام دے دیا اور details کے لئے Stop دے دیا جس سے ہمیں پتا چلے گا کہ اس سے فنکشن بند ہوگا۔

```
<script>
```

```
function myFunction() {
```



```
alert("Hello");
}
```

اس کے بعد ہم نے ایک فنکشن بنایا myFunction کے نام سے اور اس میں ایک alert کا استعمال کیا ہے۔ اس سے اب یہ ہوگا کہ جب کوئی یوزر ہمارے Start کے بٹن پر کلک کرے گا تو اسے ہر 3 سیکنڈز کے بعد ایک الرٹ ملے گا جب تک کہ وہ Stop پر کلک نہ کر دے۔

اس کا رزلٹ یہ ہوگا۔

کلک کرنے کے بعد	کلک کرنے سے پہلے
	

setInterval کو ہم clock کے باب میں پڑھیں گے۔

Clock

ہم یہاں آپ کو دو حصوں میں گھڑی بنانا سکھائیں گے۔ پہلے میں ہم بنیادی چیزوں کے بارے میں پڑھیں گے اور سیکھیں گے۔ اور دوسرے میں ہم اس گھڑی کو چلانا سکھیں گے۔ جس کے لئے ہم CSS اور HTML کا بھی استعمال کریں گے۔

گھڑی نمبر 1

```
var time = new Date ();
```



```

var hour = time.getHours();

var min = time.getMinutes();

var sec = time.getSeconds();

document.write ("Current time is: "+"<b>" + hour + ":" + min + ":" + sec + "</b>");

</script>

```

اس گھڑی میں ہم نے سب سے پہلے ایک ویریبیل بنایا time کے نام سے اور اس کو ویلو دے دی () new Date کی یہ ہماری گھڑی کو وہ تمام چیزیں دے گا جن کی ہمیں ضرورت ہے۔ اس کے بعد ہم نے ایک اور ویریبیل بنایا hour کے نام سے اور اس کو time کے ویریبیل کے ساتھ concatenate کر کے getHours() کا فنکشن دے دیا۔ یہ ہماری گھڑی کو گھنٹے دے گا۔ اس کے بعد ہم نے ایک اور ویریبیل بنایا min کے نام سے اور اس کو بھی concatenate کر دیا time سے اور اس کو بھی ویلو دے دی getMinutes() کے فنکشن کی جس سے یہ ہمیں اپنی گھڑی میں منٹس دے گا۔ اس کے بعد ہم نے ایک اور ویریبیل بنایا sec کے نام سے اور اس میں بھی وہی طریقہ کار استعمال کر کے getSeconds() کے فنکشن کو بطور ویلو دے دیا جس سے اب ہمیں ہماری گھڑی میں سیکنڈز بھی مل گئے۔

آخر میں ہم نے ان سب کو document.write() کے اندر پرنٹ کر دیا۔ یہاں اب آپ دیکھ سکتے ہیں کہ ہم نے
 کا ٹیگ بھی استعمال کیا ہے document.write() کے اندر اور ایک اسٹرنگ ویلو بھی دی ہے "Current time is: " کی اور ان سب کو جوڑنے کے لئے ہم نے + کے سائن کا استعمال کیا ہے یہ ہم نے اس لئے کی ہے کہ ہماری گھڑی کا وقت bold نظر آئے۔

اس کا رزلٹ یہ ہے۔

Current time is: 5:44:1

لیکن اس میں آپ دیکھ سکتے ہیں کہ سیکنڈز 1 سے شروع ہو رہے ہیں اور اس کے آگے AM یا PM بھی نظر نہیں آ رہا کیونکہ یہ اس وقت 12 گھنٹے کی گھڑی ہے اور ڈیفالٹ طور پر JavaScript کے اندر 0 نہیں آتا اس کام کے لئے گھڑی نمبر 2 دیکھیں۔ اور ابھی یہ چلے گی بھی نہیں ہمیں پیج کو بار بار refresh کرنا پڑے گا۔

```

var time = new Date ();
var hour = time.getHours();
var min = time.getMinutes();
var sec = time.getSeconds();

```

یہاں آپ دیکھ سکتے ہیں کہ یہ کوڈ اوپر والا ہی ہے۔ ہمیں بالکل ایسا ہی لکھنا ہے۔

```

document.write ("Current time is: "+"<b>" +hour+": "+min+": " + " sec "+"
"+sort+"</b>");

```

لیکن یہاں ہم نے کوڈ میں sort کے نام سے ایک ویریل کا نام لکھا ہے یہ AM اور PM کے لئے ہے۔ نیچے کوڈ دیکھیں۔

```

if (hour<12){
    sort="AM";
} else{sort="PM";}

```

یہاں ہم نے سب سے پہلے ایک if کے فنکشن کے اندر مائٹ دی کہ اگر hour کا ویریل less than 12 ہو تو sort کے ویریل کے اندر اس کی ویلو کرو AM اور else کے اندر sort کی ویلو PM کر دو۔ اب اس سے ہوگا یہ کہ جب ہمارے گھنٹے 12 سے ہم ہو گئے تو یہ AM دے گا اور جب وہ 12 گھنٹوں سے اوپر جائیں گے یعنی جب 13 بجے گیس تو PM ہوگا۔

```

if(hour>12){
    hour=hour-12;
}

```

اب یہاں آپ دیکھ سکتے ہیں کہ ہم نے if کے فنانکشن کو دوبارہ استعمال کرتے ہوئے مائنڈ دی کے کہ hour کا ویریبل زیادہ ہو 12 سے تو hour کے ویریبل کے اندر ہی hour کے ویریبل میں 12 کو مائنس کر دو۔

نوٹ: اس کوڈ میں پہلے والا hour کا ویریبل اس کی ویلو والے hour سے مختلف ہے اور پہلے والا ہی گنا جائے گا۔
اب اس کوڈ سے ہو گا یہ کہ جیسے ہی ہماری گھڑی میں 13 بجے کا وقت ہو گا وہ پھر سے 1 سے شروع ہو جائے گا۔

```
if (min<10){
min="0"+min;
}
```

اب اگلے کوڈ کی طرف آتے ہیں اس کوڈ میں ہم نے if کے ویریبل کے اندر مائنڈ دی کہ اگر min کا ویریبل کم ہو 10 سے تو min کے ویریبل کی ویلو 0 کر دو اور پھر min کا ویریبل ہی اس میں دوبارہ جمع کر دو۔ اس سے ہو گا یہ کہ جیسے ہی منٹس 1 سے شروع ہونگے 1 کے آگے 0 لگ جائے گا جو 9 تک چلے گا اور دس پھر خود بہ خود ختم ہو جائے گا۔

```
if (sec<10){
sec="0"+sec;
}
```

سیکنڈز کے لئے بھی ہم نے min کا ہی طریقہ استعمال کے کیا ہے۔ اس سے بھی جب سیکنڈز 1 سے شروع ہونگے تو 1 کے آگے 0 آجائے گا اور 9 کے بعد خود بہ خود ختم ہو جائے گا۔

اس گھڑی کا رزلٹ یہ ہو گا۔

6:02:02 AM

لیکن ابھی بھی اس کو refresh کرنا پڑے گا چلانے کے لئے۔ اب اس کو سمجھنے کے لئے ہمیں ایڈوانس گھڑی میں جانا ہوگا۔

ایڈوانس گھڑی نمبر 1

یہاں سب سے پہلے ایک فنکشن بنایا myTime() کے نام سے اور اس میں document.write() کا استعمال کیا اور اپنی تیار کی ہوئی گھڑی کی تمام چیزیں اس میں لکھ دی اور ریسیٹ کو باہر لکھ دیا۔ اور if else والے تمام فنکشنز myTime() کے فنکشن کے اندر ہی لکھ دیے۔

```
var time = new Date ();
```

```
var hour = time.getHours();
```

```
var min = time.getMinutes();
```

```
var sec = time.getSeconds();
```

```
function myTime() {
```

```
document.write ("Current time is: "+"<b>" + hour + ": " + min + ": " + sec + "  
" + sort + "</b>");
```

```
if (hour < 12){
```

```
sort = "AM";
```

```
} else { sort = "PM"; }
```

```
if (hour > 12){
```

```
hour = hour - 12;
```

```
}
```

```
if (min < 10){
```



```
min="0"+min;
```

```
}
```

```
if (sec<10){
```

```
sec="0"+sec;
```

```
}
```

```
}
```

اس کے بعد ہم نے ایک ویریبیل بنایا myVar کے نام سے اور اس میں setInterval() کے فنکشن کا استعمال کریں گے۔ اس میں ہم نے فنکشن لکھ کر اس کا نام دے دیا myTime() اور 1000 ملی سیکنڈ زدے دیا جس سے اب یہ ہر 1 سیکنڈ میں چلے گا۔

```
var myVar = setInterval(function(){myTime()}, 1000);
```

اس کارز لٹ یہ ہوگا۔

6:45:33 AM

ایڈوانس گھڑی نمبر 2

```
<div id="clock"></div>
```

اس گھڑی میں ہم نے سب سے پہلے ایک div کا ٹیگ بنایا اور اس کو id دے دی clock کی اور اب اسی id پر ہم اپنا JavaScript اور CSS کا کام کریں گے۔ JavaScript کا کوڈ دیکھیں۔

```
<script>
```

```
var myVar = setInterval(function(){myTime()}, 1000);
```

```
function myTime() {
    var d = new Date();
    document.getElementById("clock").innerHTML =
        d.toLocaleTimeString();
}
</script>
```

سب سے پہلے ہم نے `setInterval()` کے فنکشن کا استعمال کیا۔ اس میں ہم نے فنکشن لکھ کر اس کا نام دے دیا `myTime()` اور 1000 ملی سیکنڈ زدے دیا جس سے اب یہ ہر 1 سیکنڈ میں چلے گا۔ اس کے بعد ہم نے ایک فنکشن بنایا `myTime()` کے نام سے اس کے اندر ہم نے ایک ویریبل بنایا `d` کے نام سے اور اس کو `new Date()` کی جس سے اب یہ ویریبل ساری انفارمیشن جو ہمیں چاہیے رکھ لے گا۔ پھر اس کے بعد ہم نے `document.getElementById()` کا استعمال کیا اور جو `id` ہم نے دی تھی `clock` کے نام سے وہ اس کے اندر لکھ دی۔ اس کے بعد ہم نے `innerHTML` کی مائنڈ کا استعمال کیا اور اس کے اندر `d` کے ویریبل کو `concatenate` کر کے `toLocaleTimeString()` کے ساتھ `connect` کر دیا جس سے اب یہ مقامی وقت حساب سے چلے گا۔

اس کا رزلٹ یہ ہوگا۔

6:45:33 AM

اس کے بعد ہم یہاں CSS کا استعمال کریں گے۔ جس سے ہماری گھڑی مزید خوبصورت اور دلکش بن جائے گی۔

CSS کا کوڈ ہم `style` کے ٹیگ کے اندر لکھتے ہیں۔

```
#clock { background:red; color:yellow; width:82px; font-family:arial;
border:2px dashed green; padding:1px; }
```

ہم نے اس کوڈ میں background کا رنگ الال رکھا ہے اس کے فونٹ کا کلمہ پیلا، اس کی لمبائی 82px، فونٹ کا اسٹائل arial رکھا ہے اس کا border ڈیش اسٹائل میں رکھا ہے وہ بھی سبز رنگ کا اور آخر میں padding 1px رکھی ہے۔

اس سے اب ہماری گھڑی کچھ اس طرح کی نظر آئے گی۔

6:49:30 AM

اسی طرح کے مزید کوڈ استعمال کر کے آپ مزید گھڑیاں بھی بنا سکتے ہیں۔

6:27:01 AM

6:47:50 AM

Calculator

ہم یہاں آپ کو ایک بنیادی Calculator بنانا سکھائیں گے۔ جس میں آپ جمع (+)، گھٹنا (-)، تفریق (÷)، جس کے سائن کے لئے ہم سلش (/) کا استعمال کرتے ہیں۔ اور ضرب (x) کر سکیں گے۔

```
<FORM NAME="Calc"> <TABLE BORDER=4> <TR> <TD>
```

```
<INPUT TYPE="text" NAME="Input" Size="16">
```

سب سے پہلے ہم نے ایک form کا ٹیگ لگایا جس کا نام ہم نے Calc رکھ دیا۔ اس کے بعد ہم نے ایک table بنایا جس کا border ہم نے 4 رکھ دیا اس کے بعد ہم نے ایک tr بنایا اور اس tr کے اندر ایک td بنایا اس td کے اندر ہم نے ایک input کا استعمال کیا ہے۔ یہاں ہم نے input کا type=text رکھا ہے جس سے اب یہ ایک text box بن گیا ہے اس کو ہم نے

نام دیا input اور اس کا سائز ہم نے 16 رکھا ہے۔

اس کے بعد ہم نے ایک br کا ٹیگ لگایا نئی لائن کے لئے جس سے ہمیں td کے اندر ایک اور لائن کا اضافہ مل جائے گا۔

</TD> </TR>

اس کے بعد ہم نے td اور td کے ٹیگ کو بند کر دیا۔

<TR><TD>

اس اگلی لائن میں ہم نے پھر سے ایک tr بنایا اور اس کے اندر ایک td بنایا اب اس کے اندر ہم اپنے calculator کے پہلے چار بٹن بنائیں گے۔

```
<INPUT TYPE="button" NAME="one" VALUE="1" OnClick="Calc.Input.value += '1' ">
<INPUT TYPE="button" NAME="two" VALUE="2" OnClick="Calc.Input.value += '2' ">
<INPUT TYPE="button" NAME="three" VALUE="3" OnClick="Calc.Input.value += '3' ">
<INPUT TYPE="button" NAME="plus" VALUE=" + " OnClick="Calc.Input.value += '+' ">
```

یہاں ہم نے پہلے لائن کے لئے 4 بٹن بنانے ہیں اس کام کے لئے ہم نے چار input فیلڈ بنائی اور اس کا نام پ button رکھ دیا اس سے اب ہمارے پاس چار button بن گئے ہیں۔ پہلے والے کو ہم نے نام دیا one کا اور اس کی ویلوی 1 جو کہ بٹن کے اوپر شو ہو گا اور ہمیں یا یوزر کو پتا چلے گا کہ یہ 1 نمبر کا بٹن ہے اس کے بعد ہم نے onclick کے Attribute میں جو ویلوی ہے اس کے لئے ہم نے Calc جو کہ form کا نام ہے اسے input جو کہ پہلی input فیلڈ کا نام ہے اس کے ساتھ concatenate کر دیا۔ اس سے اب یہ form کی ویلوی کو input box کی ویلوی کے ساتھ ملا دے گا۔ اس سے ہمیں calculator میں لکھا جانے والا سب کچھ پہلی فیلڈ میں شو ہو گا۔ اس کے بعد ہم نے Calc.Input کو JavaScript کے ایک نمائندہ value کے ساتھ

concatenate کر دیا اس سے اب یہ اس بٹن کی ویلوی کو اٹھا کر Calc.Input کے ساتھ ملا دے گا اور پھر ہم نے اسے + اور اس

کے بعد = کا سائن لگا کر 1 کی ویلوی دے دی جسے ہم نے اسٹرنگ کے طور پر لکھا ہے۔ + کے فوراً بعد = کے سائن کو سے ہم

Assignment اوپر پڑکتے ہیں۔ اس سے ویلوی میں پہلے ایک کا اضافہ ہو جاتا ہے۔ اس کے بعد ہم نے اسٹرنگ ویلوی 1 دی ہے ڈبل

کوئس کے ساتھ 1 اسٹرنگ ویلو ہے۔

اس کو سمجھنے میں مشکل ہو سکتی ہے اسی لئے کوشش ضروری ہے۔ یہی عمل ہم نے باقی تینوں میں بھی دہرایا ہے۔

اس کے بعد ہم نے ایک br کا ٹیگ لگایا نئی لائن کے لئے جس سے ہمیں td کے اندر ایک اور لائن کا اضافہ مل جائے گا۔ جہاں ہم اپنے اگلے چار بٹن بنائیں گے۔

```
<INPUT TYPE="button" NAME="four" VALUE=" 4 " OnClick="Calc.Input.value += '4' ">
<INPUT TYPE="button" NAME="five" VALUE=" 5 " OnClick="Calc.Input.value += '5' ">
<INPUT TYPE="button" NAME="six" VALUE=" 6 " OnClick="Calc.Input.value += '6' ">
<INPUT TYPE="button" NAME="minus" VALUE=" - " OnClick="Calc.Input.value += ' - ' ">
```

اس اگلی لائن میں ہم نے مزید چار بٹن بنائیں ہے اور یہاں بھی ہم نے وہی طریقہ کار استعمال کیا ہے جو پہلے چار بٹن بنانے کے لئے استعمال کیا تھا۔

اس کے بعد ہم نے پھر ایک br کا ٹیگ لگایا نئی لائن کے لئے جس سے ہمیں td کے اندر ایک اور لائن کا اضافہ مل جائے گا۔ اور اب ہم یہاں اپنے مزید چار بٹن بنائیں گے۔

```
<INPUT TYPE="button" NAME="seven" VALUE=" 7 " OnClick="Calc.Input.value += '7' ">
<INPUT TYPE="button" NAME="eight" VALUE=" 8 " OnClick="Calc.Input.value += '8' ">
<INPUT TYPE="button" NAME="nine" VALUE=" 9 " OnClick="Calc.Input.value += '9' ">
<INPUT TYPE="button" NAME="times" VALUE=" x " OnClick="Calc.Input.value += ' * ' ">
```

اس لائن میں بھی ہم نے وہی طریقہ کار استعمال کیا ہے اور اس مرتبہ ہم نے آخری لائن میں ضرب کے لئے * کا استعمال کیا ہے لیکن ویلو x کی دی ہے کیونکہ یہ ہمیں نظر آئے گا اور اس کا مزید کوئی کام نہیں ہے۔

اس کے بعد ہم نے ایک br کا ٹیگ لگایا نئی لائن کے لئے جس سے ہمیں td کے اندر ایک اور لائن کا اضافہ مل جائے گا۔ اور یہاں ہم اپنے آخری چار بٹن بنائیں گے اور اپنا کام مکمل کریں گے۔

```
<INPUT TYPE="button" NAME="clear" VALUE=" c " OnClick="Calc.Input.value = ' ' ">
<INPUT TYPE="button" NAME="zero" VALUE=" 0 " OnClick="Calc.Input.value += '0' ">
<INPUT TYPE="button" NAME="DoIt" VALUE=" = " OnClick="Calc.Input.value =
eval(Calc.Input.value)">
```

اس لائن میں ہم نے clear کے بٹن کو خالی رکھا ہے کیونکہ جب اس بٹن کو دبایا جائے گا تو سب کچھ ختم ہو جانا چاہیے۔ اور زیرو کے لئے ویلو زیرو اور آخر میں doit کے نام سے ایک تیسرا بٹن جو کہ = کے لئے استعمال ہوگا۔ اور اسی کوڈ کے اندر ہم eval() کا استعمال کریں گے یہ math equations کو جمع کرنے کا کام کرتا ہے۔ اسی فنکشن کے اندر ہم Calc.input.value کو ایڈ کریں گے جس سے اب جتنے بھی بٹن دبائے پر ویلو ملے وہ اس کے اندر خود بہ خود ایڈ ہو جائے گی۔ اور = کا بٹن دبائے پر ہمیں جواب مل جائے گا۔

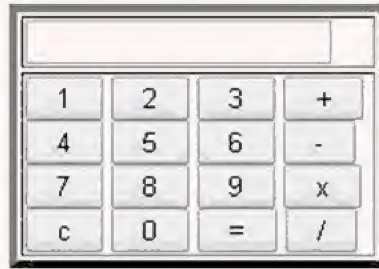
```
<INPUT TYPE="button" NAME="div" VALUE=" / " OnClick="Calc.Input.value += ' / ' ">
```

اوپر ہم نے تین بٹن بنائے تھے لیکن ابھی چوتھا بٹن رہ گیا ہے اور وہ ہے تقسیم کا یعنی / کا اس کے لئے ہم نے آخری بٹن بھی اسی طریقے کار سے تیار کیا اور پھر ہمیں تمام ٹیگوں کو آخر میں بند کرنا ہے۔ یہاں یہ بتانا چلوں کہ input کا ٹیگ بند نہیں ہوتا۔

</TD></TR></TABLE></FORM>

اس کے بعد ہم نے آخر میں تمام ٹیگوں table، tr، td اور form سب کو بند کر دیا۔ اب ہمارا Calculator تیار ہے۔

اس کارزلٹ یہ ہوگا۔



ہم اسے CSS کا استعمال کر کے مزید خوبصورت بھی بنا سکتے ہیں۔



Temperature Convertor

ہم یہاں آپ کو ایک بنیادی Temperature Convertor بنانا سکھائیں گے۔ جس میں ہم Fahrenheit اور Celsius کو ایک دوسرے سے Convert کریں گے۔ اس کام کے لئے ہم Math.round() کا استعمال کریں گے۔ لیکن اس سے پہلے ہم شروع کریں ہمیں یہ پتا ہونا چاہئے ایک Fahrenheit کتنے Celsius کا ہوتا ہے یا ایک Celsius کتنے Fahrenheit کا ہوتا ہے۔

1 Fahrenheit = -17.2222 Clesius
1 Clesius = 33.8 Fahrenheit

```
<input id="c" onkeyup="convert('C')">Celsius</br></br>
```

```
<input id="f" onkeyup="convert('F')">Fahrenheit
```

یہاں ہم نے دو input box بنائے ایک کو id دی c کی اور دوسرے کو f کی دونوں میں ہی ہم نے onkeyup کے event کا استعمال کیا ہے۔ اور دونوں میں convert کے نام سے جو فنکشن بنایا ہے اس کا نام دے دیا۔ اور دونوں کو f اور c کے جو ویریبلز ہم نے convert کے فنکشن کے اندر بنائے ہیں وہ دے دے۔

```
<script>
function convert(degree) {
    if (degree == "C") {

F = document.getElementById("c").value * 9 / 5 + 32;

document.getElementById("f").value = Math.round(F);

    } else

C = (document.getElementById("f").value - 32) * 5 / 9;

document.getElementById("c").value = Math.round(C);

    }

}

</script>
```

اب یہاں ہم نے script کے ٹیگ کے اندر ایک فنکشن بنایا convert کے نام سے اور اس کے اندر ایک undefined ویریبل بنایا degree کے نام سے پھر ہم نے فنکشن کے اندر ایک if کا فنکشن استعمال کیا اور اس کے اندر پروپٹی دی کہ اگر degree کا ویریبل

برابر کو C کے تو اور اس کو کنڈیشن دینے کے لئے ہم نے پہلے F کے نام سے ایک ویریبیل بنایا اور اس کو ویلو دے دی
 document.getElementById اور جو id ہم نے پہلے input فیلڈ میں دی تھی C کے نام سے وہی دے دیا اور پھر ہم نے
 value کی مائنڈ کے ساتھ اسے concatenate کر دیا جس سے اب یہ اس میں سے ویلو اٹھالے گا اور اسے بعد ہم نے
 9/5+32 کو value سے ضرب کر دیا۔ اس طرح ضرب ہونے کے بعد جو ویلو ملے گی وہ ہمیں کنورٹ شدہ ویلو ملے گی۔ اگلی میں ہم
 نے پھر سے document.getElementById کا استعمال کیا اور اس کے اندر جو ہم نے دوسرے بٹن میں f کی id بنائی تھی
 اسے لکھا اور اسے بھی value کی مائنڈ کے ساتھ concatenate کر دیا پھر ہم نے ایک اور میٹھ
 او بجیکٹ Math.round() کا استعمال کیا اس سے جو قدرتی ویلو ہوتی ہے وہ ہمیں ملتی ہے۔ اس کے اندر ہم نے F کے ویریبیل کو ایڈ
 کر دیا۔

اس کے بعد ہم نے else کے لئے بھی وہی طریقہ کار استعمال کرتے ہوئے پہلے C کا ویریبیل بنایا اس کے اندر ہم نے
 document.getElementById کا استعمال کیا لیکن اس بار ہم نے اسے رائونڈ بریکٹس کے اندر لکھا ہے کیوں کہ اس بار ہمیں
 یہ تمام کی تمام ویلو ضرب کرنی ہے۔ ہم نے document.getElementById کو جو ہم نے دوسرے بٹن میں id بنائی
 تھی f کے نام سے وہ دے دی اور value کی مائنڈ کے ساتھ concatenate کر دیا۔ اسے بعد 32- کو اگلا کر رائونڈ بریکٹ کو
 بند کیا اور اس کو ضرب کر دیا 5 / 5 سے اس کے بعد ہم نے پہا طریقہ ہی استعمال کرتے ہوئے
 document.getElementById کا استعمال کیا اور اس میں پہلے والی C کی id لکھ دی اور اس کو value کی مائنڈ کے
 ساتھ concatenate کر دیا اور اس میں بھی Math.round() کے او بجیکٹ کا استعمال کیا لیکن اس بار اس کے اندر ہم نے
 F کے ویریبیل کی جگہ C کے ویریبیل کو ایک کر دیا۔

اب اس سے ہو گا یہ کہ جب بھی کوئی یوزر پہلے input فیلڈ کے اندر کوئی دیا داخل کرے گا تو وہ ویلو پہلے یا دوسرے جس میں یوزر ویلو
 داخل کرتا ہے اس سے اٹھا کر پہلے ضرب پھر تقسیم ہوگی اور آخری میں چونکہ ہم نے onkeyup کے Attribute کا استعمال کیا ہے
 اس لئے onclick کے برعکس یہ ہمیں فوراً ہی رزلٹ دے دیگا۔
 اس کا رزلٹ یہ ہوگا۔

 Celsius

 Fahrenheit